

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΛΟΓΙΣΜΙΚΟΥ  
ΧΑΜΗΛΟΥ ΕΠΙΠΕΔΟΥ ΓΙΑ ΜΕΤΑΓΩΓΕΙΣ ΠΑΚΕΤΩΝ  
ΥΨΗΛΩΝ ΤΑΧΥΤΗΤΩΝ**

***ΧΡΗΣΤΟΣ Ζ. ΛΟΛΑΣ***

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΗΡΑΚΛΕΙΟ ΝΟΕΜΒΡΙΟΣ 2001**

## Πρόλογος

Στο παρόν κείμενο περιγράφονται τα πεπραγμένα της μεταπτυχιακής εργασίας μου που έλαβε χώρα στο Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης και στο Ινστιτούτο Πληροφορικής του Ι.Τ.Ε. Η παραπάνω εργασία πραγματοποιήθηκε στα πλαίσια του έργου Ε.Π.Ε.Τ. «Δίκτυα Πρόσβασης Ολοκληρωμένων Υπηρεσιών (ΔΠΟΛΟ)», στο οποίο συμμετείχαν το Π.Κ. και το Ι.Π./Ι.Τ.Ε. σε συνεργασία με το Ε.Μ.Π. και την Intracom.

Το περιεχόμενο της παρούσας μεταπτυχιακής εργασίας αφορά τον σχεδιασμό και την υλοποίηση λογισμικού χαμηλού επιπέδου για μία κάρτα Εξυπηρετητή Κίνησης ABR η οποία αποτελεί υποσύστημα του παραπάνω έργου και υλοποιήθηκε στο Πανεπιστήμιο Κρήτης και το Ι.Τ.Ε.

### Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον επόπτη μου σε αυτή την εργασία καθ. Μ. Κατεβαΐνη, για την ανάθεση της, την υποστήριξη και την συνεχή συνεργασία που μου παρείχε. Επίσης θα ήθελα να ευχαριστήσω, τον καθ. Δ. Πνευματικάτο για την συνεργασία και τις καίριες συμβουλές, τον καθ. Δ. Σερπάνο για την συμβολή του στα πρώτα στάδια του έργου, τα μέλη του εργαστηρίου Αρχιτεκτονικής Υπολογιστών και VLSI, Γ. Καλοκαιρινό και Μ. Λυγεράκη για τον χρόνο που μου αφιέρωσαν στις δύσκολες στιγμές και την απο κοινού επίλυση πολλών προβλημάτων καθώς και όλους τους συμφοιτητές μου και ιδιαίτερα τους Δ. Καψάλη και Γ. Παπαδάκη για την συνεργασία κατά την διάρκεια του έργου αυτού.

Τέλος θα ήθελα να ευχαριστήσω την Γ.Γ.Ε.Τ., το Πανεπιστήμιο Κρήτης και το Ι.Τ.Ε. για την οικονομική στήριξη του έργου ΔΠΟΛΟ και μέσω αυτού των σπουδών μου.

Χρήστος Λόλας

## Περιεχόμενα

<b>Εισαγωγή</b>	<b>6</b>
<b>Συνεισφορά της εργασίας</b>	<b>7</b>
<b>1 Κόμβος Πρόσβασης</b>	<b>7</b>
1.1 Γενικά χαρακτηριστικά	7
1.2 Απαιτήσεις συστήματος	8
1.3 Λειτουργία	9
1.3.1 Το Επίπεδο Χρήστη (User Plane)	9
1.3.2 Επίπεδο Ελέγχου (Control Plane)	9
1.3.3 Επίπεδο Διαχείρισης (Management Plane)	10
1.4 Περιγραφή της γενικής αρχιτεκτονικής του κόμβου πρόσβασης	11
1.4.1 Δομή	11
1.4.2 Περιγραφή Υποσυστημάτων	12
1.4.3 Περιγραφή δικτύου διασύνδεσης	14
1.5 Οργάνωση και Λειτουργικότητα του συστήματος	18
1.5.1 Λειτουργικότητα Επιπέδου Χρήστη	19
1.5.2 Λειτουργικότητα Επιπέδου Ελέγχου	21
1.5.3 Λειτουργίες Επιπέδου Διαχείρισης	23
1.5.4 Δρομολόγηση δεδομένων στο εσωτερικό του AN-Switch	25
1.6 Εσωτερική επικοινωνία του AN-Switch	25
1.6.1 Είδη μηνυμάτων εσωτερικής επικοινωνίας	25
1.6.2 Φυσικό επίπεδο μεταφοράς μηνυμάτων	26
1.6.3 Περιγραφή πρωτοκόλλου εσωτερικής επικοινωνίας	27
<b>2 Ο Εξυπηρετητής ABR κίνησης</b>	<b>38</b>
2.1 Δομή και Λειτουργικότητα της κάρτας Εξυπηρετητή ABR	38
2.2 Αρχιτεκτονική της κάρτας Εξυπηρετητή ABR	41
2.2.1 Το στοιχείο μεταγωγής	42
2.2.2 Η Μονάδα Εξυπρέτης Κίνησης ABR (M.E.K.A.)	42
2.2.3 Ο Επεξεργαστής Κελιών	43
2.2.4 Εσωτερικός Επεξεργαστής Ελέγχου	44
<b>3 Το λογισμικό της εξυπηρετητή ABR</b>	<b>45</b>
3.1 Περιγραφή Λειτουργίας	45
3.1.1 Διαχείριση	46
3.1.2 Έλεγχος Ροής	47
3.2 Αρχιτεκτονική Λογισμικού της ABR κάρτας	47
3.2.1 Χαμηλό Επίπεδο	48
3.2.2 Μεσαίο Επίπεδο	53
3.2.3 Υψηλό Επίπεδο	55
3.3 Σχεδίαση Λογισμικού	57
3.3.1 Το Λειτουργικό Σύστημα (VxWorks)	59
3.3.2 Διεπαφές Χειρισμού Φυσικών Στοιχείων (Δ.Χ.Φ.Σ.)	62
3.3.3 Οδηγός AAL	65
3.3.4 Εσωτερικό Πρωτόκολλο Επικοινωνίας (ICP)	71

3.3.5	Οντότητες Διαχείρισης	72
3.3.6	Βιβλιοθήκη RA (RA Library)	78
3.3.7	Εφαρμογή Ελέγχου Ροής	80
3.3.8	Οργάνωση και Ολοκλήρωση των οντοτήτων λογισμικού	84
<b>3.4</b>	<b>Υλοποίηση λογισμικού</b>	<b>91</b>
3.4.1	Περιγραφή κώδικα	91
3.4.2	Εργαλεία	92
3.4.3	Εγκατάσταση	92
<b>4</b>	<b>Εγκατάσταση, Έλεγχος και Μετρήσεις του συστήματος</b>	<b>93</b>
4.1	Έλεγχος του Υλικού της κάρτας	93
4.2	Εγκατάσταση Λογισμικού	95
4.2.1	Εγκατάσταση λειτουργικού συστήματος	95
4.2.2	Εγκατάσταση εφαρμογών λογισμικού	96
4.3	Έλεγχος Λογισμικού	98
4.4	Μετρήσεις-Αποτελέσματα	101
4.4.1	Λειτουργία Διαχείρισης	101
4.4.2	Λειτουργία Ελέγχου Ροής	102
<b>5</b>	<b>Συμπεράσματα – Προτάσεις - Επισκόπηση</b>	<b>104</b>
5.1	Συμπεράσματα	104
5.2	Επισκόπηση εργασιών	106
5.3	Προτάσεις για μελλοντική εργασία	107
	Ο Προγραμματιστής Cells (Cell Scheduler)	109
	Διεπαφή Μ.Ε.Κ.Α. με τον εσωτερικό επεξεργαστή ελέγχου.	110
<b>Παραρτήματα</b>		
<b>Βιβλιογραφία</b>		<b>129</b>

## Πίνακας Εικόνων

Εικόνα 1 Διεπαφές Δικτύου.....	10
Εικόνα 2 Δομή του Κόμβου Πρόσβασης.....	11
Εικόνα 3 Διαμοιραζόμενος διάυλος CellBus.....	14
Εικόνα 4 Πλαίσιο μετάδοσης στο CellBus.....	15
Εικόνα 5 Μορφή πλαισίων του διαύλου σε λειτουργία 16 και 32 μονάδων.....	17
Εικόνα 6 Μορφή επικεφαλίδας στο διάυλο CellBus.....	18
Εικόνα 7 Δομή και τοποθέτηση των υποσυστημάτων του Κόμβου Πρόσβασης.....	19
Εικόνα 8 Ροές Δεδομένων Επιπέδου Χρήστη.....	20
Εικόνα 9 Ροές ελέγχου.....	23
Εικόνα 10 Η θέση του ICP στην αρχιτεκτονική λογισμικού του AN-Switch.....	27
Εικόνα 11 Οι διεπαφές του ICP.....	28
Εικόνα 12 Η κωδικοποίηση των μηνυμάτων στο ICPIF.....	29
Εικόνα 13 Η μορφή του μηνύματος ICP_OPEN_CONNECTION.....	31
Εικόνα 14 Η μορφή του μηνύματος ICP_CLOSE_CONNECTION.....	31
Εικόνα 15 Η μορφή του μηνύματος ICP_OPEN_DATA_CHAN.....	33
Εικόνα 16 Η μορφή το μηνύματος ICP_DEL_DATA_CHAN.....	34
Εικόνα 17 Η κωδικοποίηση των μηνυμάτων στο ICPSIF.....	34
Εικόνα 18 Μορφή του μηνύματος ICP_OPEN_DATA_CHAN στο ICPSIF.....	36
Εικόνα 19 Η μορφή του μηνύματος ICP_DEL_DATA_CHAN στο ICPSIF.....	36
Εικόνα 20 Η υλοποίηση των λειτουργιών από τα φυσικά στοιχεία της ABR κάρτας.....	39
Εικόνα 21 Ροή Κίνησης στο εσωτερικό της κάρτας Εξυπηρετητή ABR.....	41
Εικόνα 22 Οργάνωση της κάρτας Εξυπηρετητή ABR.....	43
Εικόνα 23 Ιεραρχικά Επίπεδα Λογισμικού κάρτας Εξυπηρετητή ABR.....	48
Εικόνα 24 Δομή του Μεσαίου Επιπέδου.....	54
Εικόνα 25 Η δομή του Υψηλού Επιπέδου και η σύνδεση του με τα χαμηλότερα επίπεδα.....	56
Εικόνα 26 Επίπεδα Σχεδίασης Λογισμικού.....	59
Εικόνα 27 Δομή και αντιστοίχιση των ΔΧΦΣ.....	63
Εικόνα 28 Αναλυτική δομή του Οδηγού AAL.....	65
Εικόνα 29 Λειτουργία και μεταφορά δεδομένων στον Οδηγό AAL.....	68
Εικόνα 30 Διάγραμμα ροής της P.X.Δ.....	69
Εικόνα 31 Διάγραμμα ροής Ρουτινών AALWrite, AALRead.....	70
Εικόνα 32 Διάγραμμα Ροής λειτουργίας ICP.....	71
Εικόνα 33 Διάγραμμα Ροής Ο.Δ.Σ.....	75
Εικόνα 34 Διάγραμμα Ροής Ο.Ε.Δ.Σ.....	76
Εικόνα 35 Λειτουργία και μεταφορά πληροφορίας Ελέγχου Ροής.....	78
Εικόνα 36 Εσωτερικοί πίνακες παραμέτρων για τον έλεγχο ροής.....	81
Εικόνα 37 Διάγραμμα Ροής Υλοποίησης Αλγορίθμου Ροής.....	82
Εικόνα 38 Οργάνωση Λογισμικού για την έκδοση που υποστηρίζει μόνο Διαχείριση (οδηγός AAL πάνω από E.K.).....	87
Εικόνα 39 Οργάνωση Λογισμικού για την έκδοση που υποστηρίζει μόνο Διαχείριση (οδηγός AAL πάνω από Σ.Μ.).....	88
Εικόνα 40 Οργάνωση Λογισμικού για την έκδοση που υποστηρίζει Διαχείριση και Έλεγχο Ροής με χειρισμό cells από το ίδιο φυσικό στοιχείο.....	89
Εικόνα 41 Οργάνωση Λογισμικού για την έκδοση που υποστηρίζει Διαχείριση και Έλεγχο Ροής με χειρισμό cells από διαφορετικά φυσικά στοιχεία.....	91
Εικόνα 42 Διάταξη για επικοινωνία του Host με την πόρτα BDM του επεξεργαστή.....	94

## Εισαγωγή

Τα τελευταία χρόνια η χρήση τηλεπικοινωνιακών συστημάτων και δικτύων παρουσιάζει αλματώδη αύξηση. Καινούργιες υπηρεσίες και προϊόντα δημιουργούνται και διακινούνται μέσω δικτύου. Η υπηρεσίες αυτές έχουν ήδη αρχίσει και αλλάζουν τον τρόπο της λειτουργίας των επιχειρήσεων όσο και την καθημερινή ζωή των ανθρώπων. Η συνεχώς αυξανόμενη ζήτηση δικτυακών υπηρεσιών και ο εκρηκτικός ρυθμός αύξησης του αριθμού των χρηστών του δικτύου έχει σαν επακόλουθο την συνεχή ζήτηση για μεγαλύτερο εύρος ζώνης και ποιότητα υπηρεσίας.

Η συνεχής αύξηση της κίνησης και των χρηστών του δικτύου έχει σαν αποτέλεσμα την εισαγωγή καινούργιων τεχνολογιών στα τηλεπικοινωνιακά δίκτυα. Τεχνολογίες όπως το ATM, xDSL, MPLS, SDH την τελευταία δεκαετία έχουν αρχίσει να χρησιμοποιούνται τόσο για τη πρόσβαση των χρηστών στο δίκτυο όσο και για τον πυρήνα του δικτύου.

Δεδομένου του κόστους ανάπτυξης οι σύγχρονες τάσεις στη σχεδίαση των μεταγωγέων υποδεικνύουν αρχιτεκτονικές με χαρακτηριστικά που περιλαμβάνουν υψηλή αποδοτικότητα, ευελιξία και αρθρωτή (modular) δομή. Ακολουθώντας αυτή την νοοτροπία ανάπτυξης οι κόμβοι και οι μεταγωγείς θα ανταποκρίνονται στις μελλοντικές τηλεπικοινωνιακές ανάγκες, οι οποίες κυρίως σχετίζονται με την αύξηση και την διαχείριση του εύρους ζώνης και την υποστήριξη νέων πρωτοκόλλων και εφαρμογών. Επιπλέον, η τεχνική σχεδίασης συστήματος με αρθρωτή αρχιτεκτονική παρέχει την δυνατότητα τροποποιήσεων και προσθήκη ή αφαίρεση λειτουργικών τμημάτων χωρίς την ανάγκη πλήρους επανασχεδίασης του συστήματος.

Η περισσότερες σχεδιάσεις μεταγωγέων χρησιμοποιούν μια κεντρική μονάδα επεξεργασίας για την υποστήριξη των διαδικασιών των πρωτοκόλλων που δεν έχουν αυστηρούς χρονικούς περιορισμούς. Η λειτουργία τους υποστηρίζεται από περιφερειακές μονάδες εξειδικευμένου υλικού που υλοποιούν την επεξεργασία που απαιτείται σε πραγματικό χρόνο. [13]

Αυτές οι σχεδιάσεις απαιτούν για την υλοποίηση τους την ανάπτυξη ενσωματωμένων (embedded) συστημάτων πάνω στα οποία θα πραγματοποιείται ο διαχωρισμός της υλοποίησης των λειτουργιών σε υλικό και λογισμικό.

Το λογισμικό που σχεδιάζεται και υλοποιείται σε τέτοια συστήματα καλύπτει ένα μεγάλο εύρος λειτουργιών, από διεπαφές χαμηλού επιπέδου με το υλικό έως και εκτέλεση πολύπλοκων αλγορίθμων σε υψηλό επίπεδο. Η υψηλή αποδοτικότητα, ευελιξία και αρθρωτή (modular) δομή πρέπει να χαρακτηρίζουν και το λογισμικό που υλοποιείται σε συνεχώς εξελισσόμενα συστήματα όπως αυτά των μεταγωγέων.

Η παρούσα εργασία αποτελεί μέρος του έργου «Δίκτυα Πρόσβασης Ολοκληρωμένων Υπηρεσιών (ΔΠΠΟΛΟ)» το οποίο είχε ως σκοπό την ανάπτυξη ενός μεταγωγέα που θα υλοποιεί έναν κόμβο πρόσβασης VDSL συνδέσεων προς το κεντρικό δίκτυο ATM. Το μέρος αυτού του έργου πάνω στο οποίο βασίστηκε αυτή η εργασία είναι η σχεδίαση και υλοποίηση μίας κάρτας Εξυπηρετητή Κίνησης ABR.

## Συνεισφορά της εργασίας

Η συνεισφορά της παρούσας μεταπτυχιακής εργασίας είναι η σχεδίαση και υλοποίηση του απαραίτητου λογισμικού λογισμικού για την εξυπηρέτηση της λειτουργικότητας της κάρτας Εξυπηρετητή Κίνησης ABR. Πιο συγκεκριμένα πραγματοποιήθηκε ο προσδιορισμός των διεπαφών υλικού λογισμικού, υλοποιήθηκαν οντότητες για την διαχείριση της κάρτας και την αποτελεσματική προσαρμογή της στο συνολικό σύστημα και υλοποιήθηκαν υψηλού επιπέδου λειτουργίες μεγάλης πολυπλοκότητας για τον έλεγχο ροής στον μεταγωγέα με σκοπό την ενσωμάτωση προχωρημένης λειτουργικότητας στο σύστημα και την αξιολόγηση της ως προς την απόδοση με βάση την χρήση των υπαρχόντων πόρων.

Η υλοποίηση του λογισμικού πραγματοποιήθηκε έτσι ώστε να ικανοποιούνται οι απαιτήσεις για ευελιξία και αρθρωτή δομή καθώς και η απαιτήσεις για αποδοτική εκτέλεση των λειτουργιών που απαιτούνται από το συνολικό σύστημα. Το λογισμικό ελέγχθηκε και μετρήθηκε πάνω στο πραγματικό σύστημα και προσδιορίστηκε ο βαθμός ικανοποίησης των παραπάνω απαιτήσεων. Επίσης προσδιορίστηκαν οι περιορισμοί που δημιουργούν οι πόροι του συστήματος ως προς την εκτέλεση λειτουργιών πραγματικού χρόνου και καθορίστηκαν οι απαραίτητες ενέργειες για την αποδοτική ενσωμάτωση τους στο σύστημα.

Η διάρθρωση του κειμένου της εργασίας έχει ως εξής:

Στο 1<sup>ο</sup> κεφάλαιο περιγράφονται η λειτουργία και η αρχιτεκτονική του Κόμβου Πρόσβασης που υλοποιήθηκε κατά την διάρκεια του έργου ΔΠΟΛΟ, υποσύστημα του οποίου αποτελεί η κάρτα και το λογισμικό που περιγράφεται στην εργασία.

Στο 2<sup>ο</sup> κεφάλαιο περιγράφεται η λειτουργικότητα και η αρχιτεκτονική της κάρτας Εξυπηρετητή ABR που υλοποιήθηκε από το Ινστιτούτο Πληροφορικής στα πλαίσια του έργου ΔΠΟΛΟ

Στα κεφάλαια 3,4 περιέχεται η κυριώς δουλειά αυτής της μεταπτυχιακής εργασίας. Συγκεκριμένα περιέχεται η λειτουργικότητα η αρχιτεκτονική, η σχεδίαση και η υλοποίηση του λογισμικού της κάρτας εξυπηρετητή ABR (κεφάλαιο 3) καθώς και διαδικασίες ελέγχου και οι μετρήσεις του (κεφάλαιο 4).

Τέλος στο κεφάλαιο 5 περιέχονται τα συμπεράσματα, οι προτάσεις για μελλοντική εργασία καθώς και η επισκόπηση των εργασιών κατά την διάρκεια της εκπόνησης της μεταπτυχιακής εργασίας.

## 1 Κόμβος Πρόσβασης

### 1.1 Γενικά χαρακτηριστικά

Ο Κόμβος Πρόσβασης είναι ένας ATM μεταγωγέας, για χρήση πρόσβασης στο δίκτυο των χρηστών (Access Node Switch, AN-Switch), βασιζόμενος σε αρχιτεκτονική διαμοιραζόμενου διαύλου. Ο μεταγωγέας υποστηρίζει Virtual Circuit Switching (VC Switch), υποστηρίζει δηλαδή πλήρως διαδικασίες σηματοδότησης κατά

την εγκατάσταση των κλήσεων και η αλλαγή της ATM επικεφαλίδας γίνεται στα πεδία VPI και VCI. Επίσης υποστηρίζει πλήρως τον χειρισμό κίνησης ABR και υλοποιεί όλα τα απαραίτητα στοιχεία προκειμένου η κίνηση ABR να αντιμετωπίζεται αποτελεσματικά.

Η τοπολογία του μεταγωγέα σε ένα ATM δίκτυο είναι στο δίκτυο πρόσβασης (Access Network). Τελικοί χρήστες διασυνδέονται μέσω του AN-Switch το οποίο συγκεντρώνει την κίνηση (traffic) των τελικών χρηστών και την προωθεί προς τον πυρήνα του δικτύου, λειτουργώντας ως συγκεντρωτής κίνησης (concentrator).

## **1.2 Απαιτήσεις συστήματος**

Το σύστημα AN-Switch παρέχει στην πλευρά των χρηστών φυσικές διεπαφές τύπου VDSL., ταχύτητας 2 Mbps, ενώ στην πλευρά του δικτύου υποστηρίζει φυσική διεπαφή STM-1 ταχύτητας 155 Mbps. Ο αριθμός φυσικών διεπαφών τύπου VDSL πρέπει να μπορεί να μεταβάλλεται ανάλογα με την απαίτηση για τον αριθμό των χρηστών σε κάθε εγκατάσταση. Η αρχιτεκτονική διαμοιραζόμενου διαύλου επιτρέπει την ύπαρξη μεταβλητού αριθμού καρτών που συνδέονται στον διαμοιραζόμενο δίαυλο. Συνεπώς η μεταβολή του αριθμού των φυσικών διεπαφών του συστήματος AN-Switch είναι εφικτή με την σύνδεση στο διαμοιραζόμενο δίαυλο καρτών που είναι εφοδιασμένες με φυσικές διεπαφές τύπου VDSL. Ο μέγιστος αριθμός φυσικών διεπαφών τύπου VDSL που απαιτούνται είναι ενενήντα έξι.

Το σύστημα AN-Switch υποστηρίζει διαδικασίες σηματοδοσίας για την εγκατάσταση και την απεγκατάσταση των κλήσεων σε πραγματικό χρόνο. Για την υλοποίηση των πρωτοκόλλων σηματοδοσίας όσο και για την επίβλεψη και διαχείριση τόσο σε επίπεδο κάρτας όσο και σε επίπεδο συστήματος απαιτείται η ύπαρξη μικροεπεξεργαστών που θα εκτελούν τις λειτουργίες αυτές. Σε τοπικό επίπεδο (επίπεδο κάρτας) απαιτείται η αρχικοποίηση και επιτήρηση του υλικού όσο και η εγγραφή και ανανέωση των δεδομένων στον τοπικό πίνακα μετάφρασης τα δεδομένα του οποίου χρησιμοποιούνται για την εισαγωγή πληροφορίας δρομολόγησης των ATM cells στο διαμοιραζόμενο δίαυλο όσο και για την αλλαγή της ATM επικεφαλίδας. Σε επίπεδο συστήματος απαιτείται η κεντρική διαχείριση των πόρων (resources), καθώς και ο έλεγχος συνέπειας (coherency) των δεδομένων στους τοπικούς πίνακες μετάφρασης.

Η διαδικασία της εγκατάστασης και απεγκατάστασης των συνδέσεων δεδομένων υποστηρίζει τόσο τις διαδικασίες της σηματοδοσίας τύπου UNI 4.0 [2] (SVC) όσο και την χρήση εξωτερικού τερματικού διαχείρισης στο οποίο ο διαχειριστής του δικτύου αναλαμβάνει την εγκατάσταση και την απεγκατάσταση συνδέσεων τύπου PVC (Permanent Virtual Connection). Ακόμα υποστηρίζεται η το πρωτόκολλο ILMI 4.0 [3] για την εγγραφή των ATM διευθύνσεων των χρηστών στον πίνακα δρομολόγησης του AN-Switch.



## 1.3 Λειτουργία

Η λειτουργικότητα του συστήματος του μεταγωγέα μπορεί να διαχωριστεί, σύμφωνα με το B-ISDN Protocol Reference Model [6], σε τρία επίπεδα:

1. το επίπεδο χρήστη (user plane)
2. το επίπεδο ελέγχου (control plane)
3. το επίπεδο διαχείρισης (management plane)

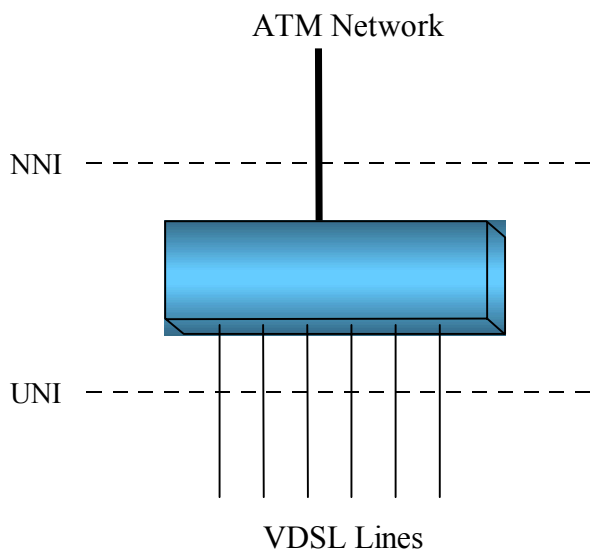
Η πληροφορία για τα τρία επίπεδα μεταφέρεται από ATM cells διαφόρων τύπων. Ο τύπος του κάθε cell καθορίζεται από την αντίστοιχη κεφαλίδα (ATM header). Η εσωτερική διαδρομή του κάθε cell καθορίζεται από πληροφορία που βρίσκεται σε επιπλέον καφαλίδες, οι οποίες προστίθενται και αφαιρούνται από το σύστημα.

### 1.3.1 Το Επίπεδο Χρήστη (User Plane)

Το επίπεδο χρήστη περιέχει τις λειτουργίες του Φυσικού επιπέδου (Physical Layer) και του Επιπέδου ATM (ATM Layer) τα οποία χειρίζονται την ροή των ATM cells που μεταφέρουν δεδομένα χρήστη (User data cells). Αυτές οι λειτουργίες αφορούν την αποδοχή και εξαγωγή cells από η προς τις ροές εισόδου / εξόδου, την επεξεργασία των κεφαλίδων (headers) αυτών των cells όπως η μετάφραση των διευθύνσεων VP/VC (VP/VC translation) και την μεταγωγή των cells από τις εισόδους του κόμβου στις κατάλληλες εξόδους.

### 1.3.2 Επίπεδο Ελέγχου (Control Plane)

Ο κόμβος πρόσβασης (AN-Switch) συνδέει τους χρήστες των γραμμών VDSL με το κεντρικό ATM δίκτυο. Για αυτόν τον σκοπό ο κόμβος βρίσκεται στο ενδιάμεσο δύο Διεπαφών Δικτύου (Network Interfaces), των UNI και NNI (**Error! Reference source not found.**). Οι χρήστες συνδέονται στον κόμβο μέσω του UNI και ο κόμβος προωθεί τα μηνύματα τους στο κεντρικό ATM Δίκτυο μέσω του NNI.



**Εικόνα 1** Διεπαφές Δικτύου

Ο Κόμβος Πρόσβασης είναι υπεύθυνος για την μετατροπή των UNI πακέτων που προέρχονται από τις γραμμές VDSL σε NNI πακέτα που προωθούνται στο κεντρικό ATM Δίκτυο μέσω γραμμών STM-1. Επίσης λειτουργεί και αντίστροφα, μετατρέποντας τα NNI πακέτα που προέρχονται από την γραμμή STM-1 σε UNI πακέτα κατάλληλα για τους χρήστες των γραμμών VDSL. Ο Κόμβος Πρόσβασης μετατρέπει επίσης εσωτερικά τα μηνύματα σηματοδότησης (signaling messages) όπως αιτήσης σύνδεσης (connection requests), επιβεβαιώσεις (acknowledgements) κ.λ.π., στην κατάλληλη μορφή (UNI ή NNI) έτσι ώστε να τα προωθήσει στον κατάλληλο προορισμό. Για αυτόν τον λόγο ο Κόμβος Πρόσβασης υλοποιεί όλα τα απαραίτητα επίπεδα ATM τα οποία απαιτούνται για τις παραπάνω λειτουργίες.

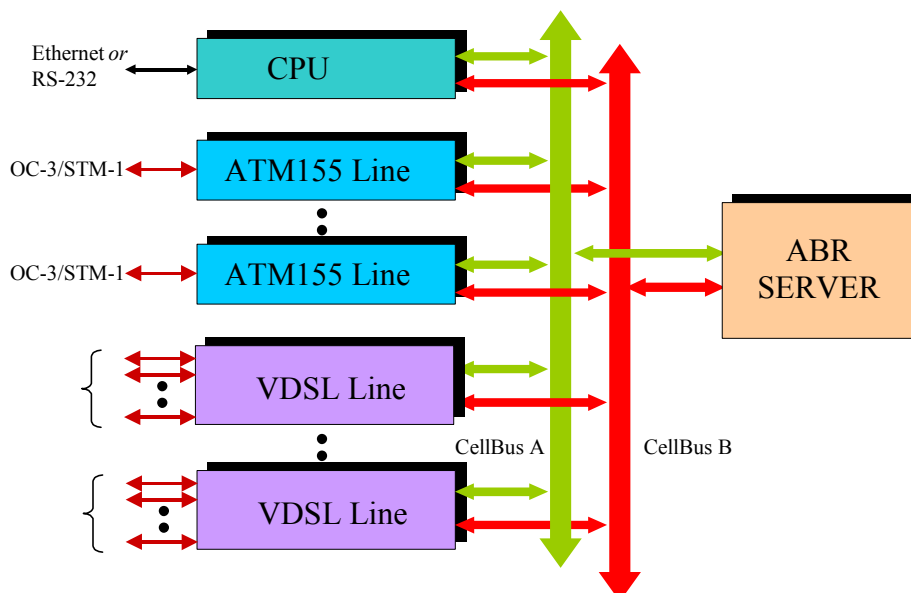
### 1.3.3 Επίπεδο Διαχείρισης (Management Plane)

Η διαχείριση του συστήματος μπορεί να διαχωριστεί σε :

1. Τοπική διαχείριση, που καλύπτει την περιοχή μίας κάρτας του συστήματος και αφορά τις αρχικοποιήσεις, τον έλεγχο και την επικοινωνία της με το υπόλοιπο σύστημα.
2. Κεντρική διαχείριση που καλύπτει την περιοχή ολόκληρου του συστήματος και αφορά τον συνολικό έλεγχο και την διακίνηση της πληροφορίας μέσα στο σύστημα.

## 1.4 Περιγραφή της γενικής αρχιτεκτονικής του κόμβου πρόσβασης

### 1.4.1 Δομή



Εικόνα 2 Δομή του Κόμβου Πρόσβασης

Η γενική αρχιτεκτονική του μεταγωγέα / κόμβου πρόσβασης φαίνεται στην Εικόνα 2:

Με βάση αυτή την αρχιτεκτονική το σύστημα θα αποτελείται από έναν αριθμό καρτών διεπαφής με τις γραμμές VDSL που συνδέουν τους χρήστες με τον κόμβο, μία ή περισσότερες κάρτα ATM 155 για την επικοινωνία του κόμβου με το κεντρικό ATM δίκτυο μέσω γραμμών OC-3/STM-1, έναν κεντρικό επεξεργαστή, έναν εξυπηρετητή ABR κίνησης και ένα κοινό δίαυλο κελιών.

Πιο συγκεκριμένα οι λειτουργίες που θα εκτελούν τα παραπάνω υποσυστήματα είναι οι ακόλουθες:

- Κοινός δίαυλος κελιών: Σε επίπεδο συστήματος, η μεταφορά των δεδομένων όσο και της πληροφορίας σηματοδότησης και διαχείρισης γίνεται μέσω συνδέσεων ATM. Η μεταγωγή των δεδομένων αυτών των συνδέσεων πραγματοποιείται μέσω της χρήσης ενός κοινού δίαυλου κελιών (Cell bus). Για λόγους ασφαλείας η αρχιτεκτονική χρησιμοποιεί δύο δίαυλους, έναν πρωτεύοντα και ένα δευτερεύοντα που θα χρησιμοποιείται μόνο σε περίπτωση αστοχίας του πρωτεύοντος.
- Κάρτα ATM155: Εκτελεί όλες τις λειτουργίες που εμπεριέχονται στο τέλος του φυσικού επιπέδου και σε ολόκληρο το επίπεδο ATM όπως τερματισμός του φυσικού επιπέδου, δρομολόγηση ATM, σηματοδότηση Q.2931, λειτουργίες OA&M, κ.τ.λ. [2][6]. Η σύνδεση πολλών τέτοιων καρτών σε ένα back plane οδηγεί στην υλοποίηση ενός μεταγωγέα.

- Εξυπηρετητής ABR κίνησης: Εκτελεί όλες τις απαραίτητες λειτουργίες για την αποδοχή, αποθήκευση και προώθηση των κελιών που μεταφέρουν την κίνηση ABR την οποία δεν μπορεί να χειριστεί η κάρτα ATM γιατί απαιτεί μεγάλο αποθηκευτικό χώρο. Η βασικότερη από αυτές τις λειτουργίες είναι η διαχείριση μεγάλου αριθμού ουρών πολλαπλών επιπέδων.
- Κεντρικός επεξεργαστής: Έχει την συνολική επίβλεψη/διαχείριση του συστήματος και θα εκτελεί λειτουργίες υψηλού επιπέδου όπως το Call Admission Control (CAC)[2].
- Κάρτα VDSL modem: Υλοποιεί την λειτουργικότητα των modem τύπου VDSL για την σύνδεση με τους χρήστες μέσω των τηλεφωνικών καλωδίων. Η οργάνωση της κάρτας αυτής είναι έξω από τον σκοπό αυτού του κειμένου.

Στην οργάνωση αυτή, η κεντριοποιημένη εξυπηρέτηση της κίνησης ABR επιτρέπει την συγκέντρωση της μνήμης προσωρινής αποθήκευσης των δεδομένων σε μία κάρτα, με συνέπεια την καλύτερη χρήση της μνήμης αυτής. Ακόμα επιτρέπει την δημιουργία συστημάτων με ή χωρίς εξυπηρετητή ABR, και αυξάνει την ευελιξία του συστήματος στην αποδοτική (από πλευράς κόστους) κάλυψη μεταβαλλόμενων αναγκών.

#### 1.4.2 Περιγραφή Υποσυστημάτων

Οι κάρτες που δομούν τον Κόμβο Πρόσβασης υλοποιούνται με όλα ή μερικά από 4 βασικά υποσυστήματα. Τα υποσυστήματα που χρησιμοποιούνται σε κάθε κάρτα καθώς και τα προϊόντα που χρησιμοποιήθηκαν για την υλοποίηση αυτών των υποσυστημάτων φαίνονται στον Πίνακας 1. Για λόγους ομοιομορφίας της αρχιτεκτονικής, σε όλες τις κάρτες χρησιμοποιήθηκαν τα ίδια προϊόντα για την υλοποίηση των υποσυστημάτων.

Υποσύστημα	Προϊόν	Κάρτα ATM	Κάρτα Εξυπηρετητή ABR	Κάρτα Κεντρικού επεξεργαστή
Στοιχείο μεταγωγής	Transwitch Cubit Pro	√	√	√
Επεξεργασία κελιών	Motorola MC92501	√	√	√
Εσωτερικού επεξεργαστή ελέγχου	Motorola MPC860	√	√	√
Φυσικού επιπέδου	PM5350	√		
Αποθήκευση ABR cells			√	

**Πίνακας 1 Υποσυστήματα του κόμβου πρόσβασης**

Οι κάρτες του συστήματος περιέχουν ένα κοινό τμήμα υλικού το οποίο αποτελείται από τον πυρήνα μεταγωγής Cubit [9], τον Επεξεργαστή Κελιών (cell processor)

MC92501 της εταιρείας Motorola [7] και τον επεξεργαστή MPC860 της εταιρείας Motorola [8]. Επίσης η κάρτα ABR περιέχει μια FPGA σχεδιασμένη στο Ινστιτούτο Πληροφορικής του ΙΤΕ. Τα ολοκληρωμένα αυτά χρησιμοποιούνται για μεταγωγή, εισαγωγή και εξαγωγή cells από την ροή δεδομένων προς και από τον επεξεργαστή, για την αλλαγή της ATM επικεφαλίδας, για τον χειρισμό των cells ABR (τοποθέτηση σε ουρά, χρονοδρομολόγηση) και αποτελούν τα κύρια υποσυστήματα υλικού του συστήματος. Στην συνέχεια περιγράφονται οι λειτουργίες που εκτελούνται από αυτά τα ολοκληρωμένα στο σύστημα AN-Switch.

Κάθε κάρτα του συστήματος είναι εφοδιασμένη με κεντρική μονάδα επεξεργασίας (CPU) MPC860 της εταιρείας Motorola. Ο επεξεργαστής στις κάρτες που διαθέτουν φυσικές διεπαφές (STM-1 και VDSL Boards) εκτελεί την αρχικοποίηση των ολοκληρωμένων της κάρτας, την επιτήρηση τους και την ανανέωση των παραμέτρων λειτουργίας τους. Τα πρωτόκολλα της ATM σηματοδοσίας εκτελούνται τοπικά στον επεξεργαστή της κάθε κάρτας. Στην CPU Board υλοποιείται ο κεντρικός έλεγχος αποδοχής κλήσεων και η διαχείριση των πόρων του συστήματος.

Ο επεξεργαστής cell (cell processor MC92501 της εταιρείας Motorola), είναι επιφορτισμένος με τις λειτουργίες αλλαγής των ATM επικεφαλίδων, εισαγωγής του CellBus Routing Header [9] στα ATM cells, και εισαγωγής/ εξαγωγής cells από/προς τον κεντρικό επεξεργαστή κάθε κάρτας. Η λειτουργία εισαγωγής/εξαγωγής cells από την ροή των δεδομένων χρησιμοποιείται για την σηματοδοσία του συστήματος, το πρωτόκολλο ILMI, και την εσωτερική επικοινωνία των καρτών με το CPU Board. Η λειτουργία εισαγωγής/εξαγωγής γίνεται σε προγραμματιζόμενες τιμές των πεδίων VPI/VCI της ATM επικεφαλίδας. Για την ATM σηματοδοσία οι τιμές αυτές είναι VPI=0 και VCI=5, και για το πρωτόκολλο ILMI οι τιμές είναι VPI=0 και VCI=16. Ο επεξεργαστής cell μπορεί να προγραμματιστεί να εκτελεί τις λειτουργίες εισαγωγής/εξαγωγής cells και για άλλες τιμές VPI/VCI σε περίπτωση μελλοντικής υλοποίησης πρωτοκόλλων που χρησιμοποιούν διαφορετικές τιμές για τα πεδία αυτά.

Η Μονάδα Εξυπηρέτησης Κίνησης ABR (FPGA / ITE) (βλ. Κεφ 2) εκτελεί λειτουργίες του επιπέδου ATM που αφορούν τον τύπο κίνησης ABR για τα cells δεδομένων. Πιο συγκεκριμένα είναι υπεύθυνη για:

- την αποθήκευση των ABR cells δεδομένων σε ουρές ανά σύνδεση (per flow queueing) μετά την είσοδό τους στον Κόμβο Πρόσβασης
- την χρονοδρομολόγηση (scheduling) των ABR cells δεδομένων κατά την έξοδό τους από τον Κόμβο Πρόσβασης ώστε να επιτυγχάνεται αποδοτική χρήση των πόρων του
- την εκτέλεση λειτουργιών ελέγχου ροής (flow control) μέσω των τεχνικών EFCI marking και Relative Rate Flow Control όπως περιγράφονται στις προδιαγραφές του ATM Forum [4].
- Την προώθηση ή αποδοχή cells Διαχείρισης και RM cells προς και από τον Επεξεργαστή Κελιών τα οποία κατευθύνονται ή προέρχονται από τον Εσωτερικό Επεξεργαστή Ελέγχου.

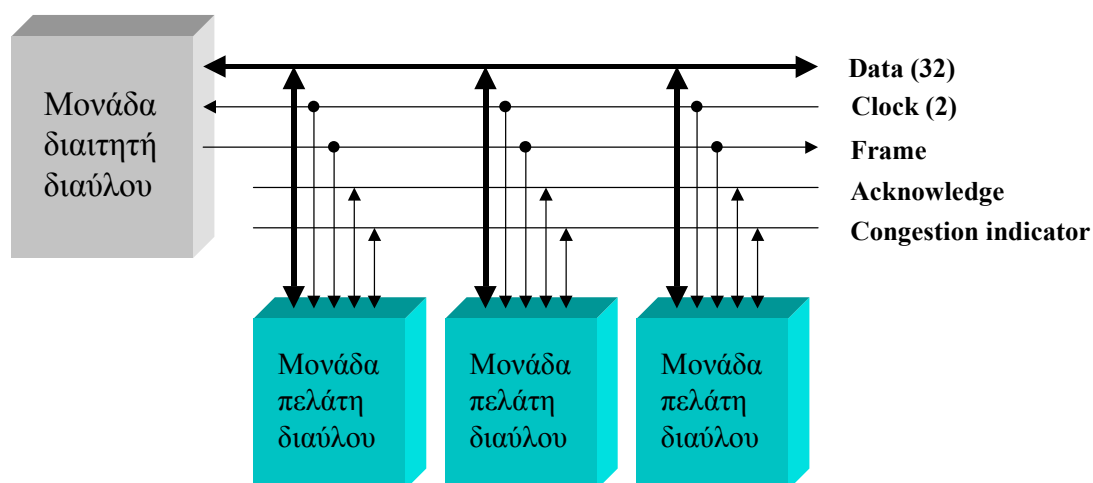
Ο πυρήνας μεταγωγής (switch fabric) που υλοποιεί την μεταγωγή των δεδομένων με χρήση του διαμοιραζόμενου διαύλου CellBus είναι το ολοκληρωμένο Cubit Pro της εταιρείας Transwitch. Ο πυρήνας μεταγωγής λαμβάνει ATM cells από τον cell processor μαζί με το κατάλληλο CellBus Routing Header που χρησιμοποιείται για την δρομολόγηση των cells στον διαμοιραζόμενο διάυλο. Κατά την παραλαβή cell από τον διαμοιραζόμενο διάυλο CellBus ο πυρήνας μεταγωγής αφαιρεί το CellBus Routing Header και παραδίδει το ATM cell στον cell processor της κάρτας προορισμού.

Οι φυσικές διεπαφές με τις οποίες είναι εφοδιασμένες οι κάρτες και υλοποιούν τις λειτουργίες του φυσικού επιπέδου όπως σκιαγράφηση των cell (cell delineation), έλεγχο λαθών, ανάκτηση και δημιουργία πλαισίων (frame recovery and generation), αποσύζευξη ρυθμού εκπομπής και μετάδοσης (cell rate decoupling).

### 1.4.3 Περιγραφή δικτύου διασύνδεσης

#### 1.4.3.1 Χαρακτηριστικά

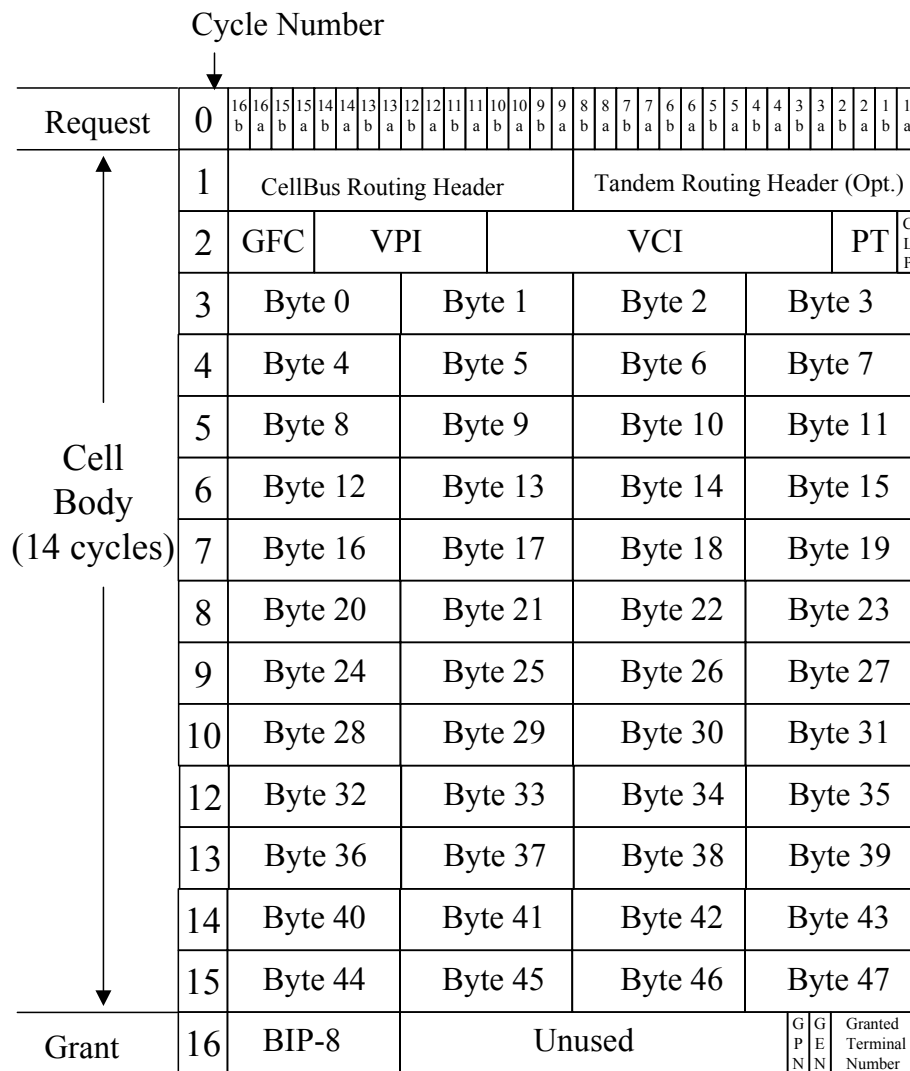
Το δίκτυο διασύνδεσης του AN-Switch βασίζεται σε αρχιτεκτονική διαμοιραζόμενου διαύλου (CellBus) [9] με κεντρική διαιτησία (arbitration) για το δικαίωμα μετάδοσης. Το δικαίωμα μετάδοσης διαρκεί συγκεκριμένο χρονικό διάστημα (χρονική σχισμή, time slot), το οποίο είναι ίσο με δεκαέξι (16) κύκλους του ρολογιού που χρονίζει την λειτουργία του διαύλου. Το χρονικό διάστημα αυτό χρησιμοποιείται για την μετάδοση ενός cell στον διάυλο CellBus. Το δίκτυο διασύνδεσης υποστηρίζει την μετάδοση σημείο προς σημείο (point to point) και πολλαπλών εκπομπών (multicasting). Το σήματα του διαύλου είναι συνολικά τριάντα επτά από τα οποία τα τριάντα δύο (32) χρησιμοποιούνται για την μεταφορά των δεδομένων. Η μέγιστη ταχύτητα λειτουργίας του διαύλου είναι τριάντα οκτώ (38) MHz, με συνολική ικανότητα διαμεταγωγής ενός (1) Gbps.



Εικόνα 3 Διαμοιραζόμενος διάυλος CellBus

Στην Εικόνα 3 απεικονίζονται τα σήματα του διαμοιραζόμενου διαύλου του AN-Switch καθώς και η συνδεσμολογία της μονάδας διαιτησίας και των μονάδων πελατών του διαύλου. Η διεπαφή τόσο των πελατών όσο και της μονάδας διαιτησίας υλοποιείται από το ολοκληρωμένο Cubit PRO της εταιρείας Transwitch [9].

### 1.4.3.2 Μετάδοση δεδομένων



**Εικόνα 4 Πλαίσιο μετάδοσης στο CellBus**

Η μετάδοση στο CellBus έχει την μορφή πλαισίου (frame) διάρκειας 16 κύκλου ρολογιού και εύρους τριάντα δύο (32) bits. Η μορφή του πλαισίου απεικονίζεται στην Εικόνα 4. Για την απόκτηση του δικαιώματος μετάδοσης στο διαμοιραζόμενο δίαυλο CellBus, οι μονάδες που θέλουν να μεταδώσουν εισάγουν κατά την διάρκεια του πρώτου κύκλου του πλαισίου ένα ή δύο bits στον δίαυλο. Ο κύκλος αυτός ονομάζεται κύκλος αιτήσεων (request cycle). Κάθε μονάδα του διαύλου μπορεί να τοποθετήσει την αίτησή της σε συγκεκριμένες θέσεις bits του διαύλου ανάλογα με το μοναδιαίο καθοριστικό τους (διεύθυνση) που την χαρακτηρίζει. Η διεύθυνση αυτή καθορίζεται από pins του ολοκληρωμένου Cubit. Η τιμή που θα τεθεί στα δυο bits που μπορεί να εισάγει μια μονάδα του διαύλου αναπαριστά την προτεραιότητα μετάδοσης που αιτείται η μονάδα αυτή. Η τιμή ένα αναπαριστά την χαμηλότερη προτεραιότητα μετάδοσης και η τιμή τρία την υψηλότερη, ενώ η τιμή μηδέν δηλώνει ότι η αντίστοιχη μονάδα δεν επιθυμεί την μετάδοση δεδομένων.

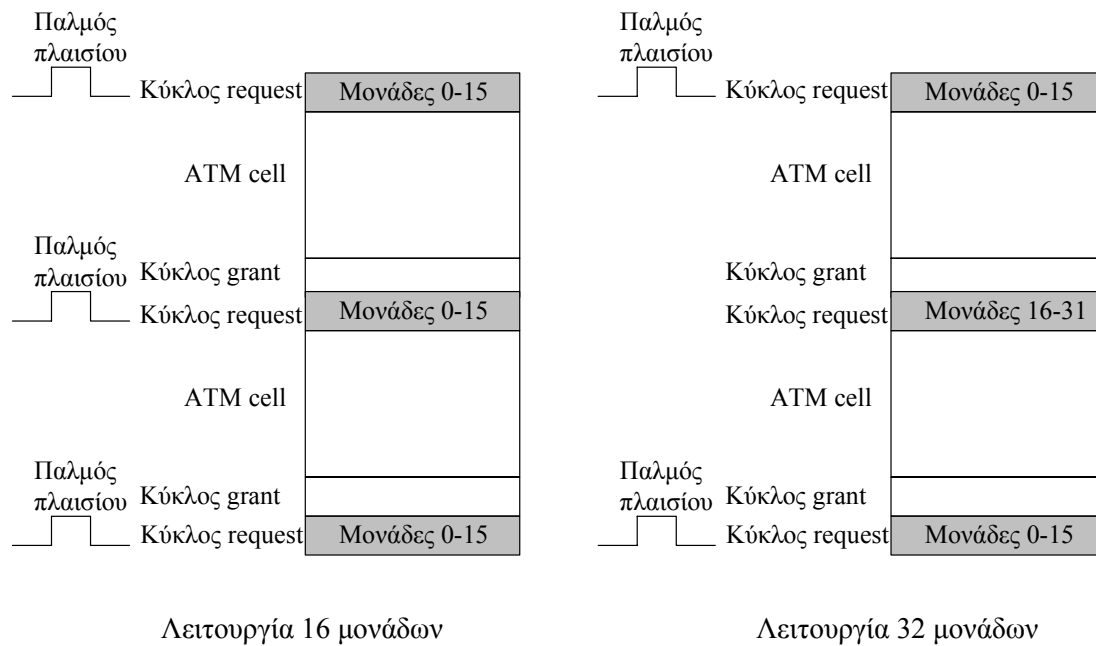
Η κεντρική μονάδα διαιτησίας λαμβάνει τις αιτήσεις μετάδοσης και εκτελεί τον αλγόριθμο διαιτησίας, σύμφωνα με τον οποίο το δικαίωμα μετάδοσης δίνεται στην μονάδα που έχει κάνει αίτηση με την υψηλότερη προτεραιότητα. Για τις αιτήσεις ίσης προτεραιότητας ακολουθείται Round Robin. Το αποτέλεσμα της διαιτησίας ανακοινώνεται κατά τον κύκλο Grant από την κεντρική μονάδα διαιτησίας. Η μονάδα που αποκτά το δικαίωμα μετάδοσης στον διαμοιραζόμενο δίαυλο θα μεταδώσει ένα cell κατά την διάρκεια του επόμενου πλαισίου.

Ο παραλήπτης του cell που μεταδίδεται καθορίζεται από την τιμή του CellBus Routing Header (Εικόνα 4) που μεταδίδεται κατά τον κύκλο ένα του πλαισίου μετάδοσης. Το cell μπορεί να έχει ένα παραλήπτη (point to point σύνδεση), ή όσες μονάδες έχουν προγραμματιστεί να λαμβάνουν cell σε συγκεκριμένα multicast groups.

### **1.4.3.3 Τρόποι λειτουργίας**

Ο δίαυλος μπορεί να υποστηρίξει τόσο την λειτουργία με δεκαέξι (16) όσο και την λειτουργία με τριάντα δύο (32) μονάδες. Η λειτουργία με δεκαέξι (16) μονάδες έχει ήδη περιγραφεί. Κατά την λειτουργία με τριάντα δύο μονάδες το πλαίσιο αντιγράφεται (duplication) ώστε να υπάρχει ένα άρτιο και ένα περιττό πλαίσιο. Η διάκριση των δύο πλαισίων γίνεται με βάση την θέση του παλμού πλαισίου (frame pulse). Οι κύκλοι μετάδοσης είναι οι ίδιοι και για τα δύο πλαίσια εκτός από τον κύκλο αιτήσεων. Ο κύκλος αιτήσεων για το άρτιο πλαίσιο συμπίπτει με τον παλμό πλαισίου, ενώ για τον περιττό κύκλο δεν μεταδίδεται παλμός πλαισίου. Κατά την διάρκεια του άρτιου πλαισίου οι μονάδες με διεύθυνση μηδέν (0) έως δεκαπέντε (15) αιτούνται για το δικαίωμα μετάδοσης, και κατά την του περιττού πλαισίου αιτούνται οι μονάδες δεκαέξι (16) έως τριάντα ένα (31). Η μορφή των πλαισίων και του παλμού πλαισίου για του δύο τρόπους λειτουργίας απεικονίζεται στην Εικόνα 5. Με αυτό τον τρόπο το συνολικό εύρος ζώνης του διαύλου χρησιμοποιείται και στην λειτουργία με τις δεκαέξι (16) μονάδες και στην λειτουργία με τις τριάντα δύο (32) μονάδες.





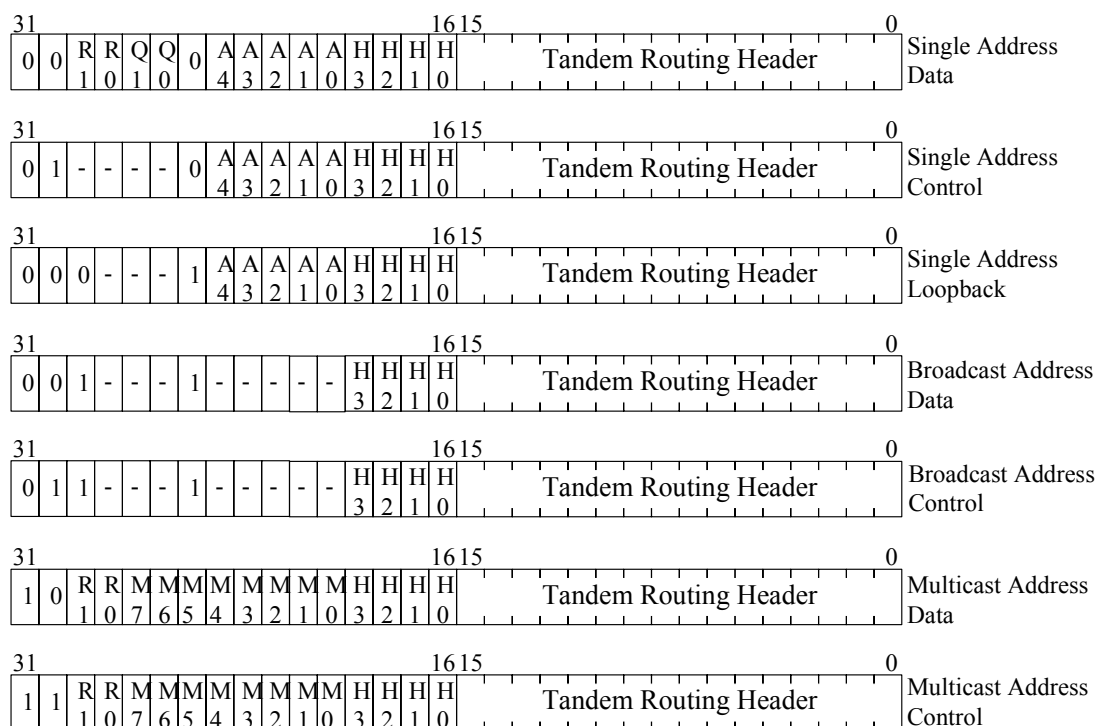
**Εικόνα 5 Μορφή πλαισίων του διαύλου σε λειτουργία 16 και 32 μονάδων**

Το πεδίο BIP-8 (Bit Interleave Parity byte) του πλαισίου μετάδοσης (Εικόνα 4) χρησιμοποιείται για την εύρεση λαθών. Το πεδίο αυτό καλύπτει από το πρώτο byte του Tandem Routing Header μέχρι το τελευταίο byte των δεδομένων (byte 47). Ο υπολογισμός του πεδίου BIP-8 γίνεται από την μονάδα μετάδοσης με επαναλαμβανόμενες πράξεις αποκλειστικού Η (XOR) στα bytes τα οποία καλύπτει. Η μονάδα που λαμβάνει τα δεδομένα κάνει τον ίδιο υπολογισμό και συγκρίνει το αποτέλεσμα με αυτό που έχει ληφθεί. Σε περίπτωση που οι τιμές δεν είναι ίσες τα δεδομένα δεν θεωρούνται έγκυρα και απορρίπτονται.

Η λειτουργία του διαύλου CellBus υποστηρίζεται από δύο σήματα ρολογιού, κοινά για όλες τις μονάδες του διαύλου. Το πρώτο σήμα ρολογιού χρησιμοποιείται για τον χρονισμό των δεδομένων κατά την εγγραφή τους στον δίαυλο, και το δεύτερο σήμα ρολογιού για την ανάγνωση των δεδομένων από αυτόν. Τα δύο σήματα ρολογιού πρέπει να είναι την ίδιας συχνότητας και να έχουν μια μικρή διαφορά φάσης για την αξιόπιστη λειτουργία του διαύλου.

#### **1.4.3.4 Δρομολόγηση στο δίκτυο διασύνδεσης**

Η μετάδοση δεδομένων στον δίαυλο CellBus μπορεί να γίνει από μια μονάδα προς μια άλλη (point to point), από μια μονάδα προς άλλες (multicasting), από μια μονάδα προς όλες (broadcasting). Για το διαχωρισμό του είδους της μετάδοσης χρησιμοποιείται το πεδίο CellBus Routing Header. Η μορφή και τα υποπεδία του CellBus Routing Header απεικονίζονται στην Εικόνα 6.



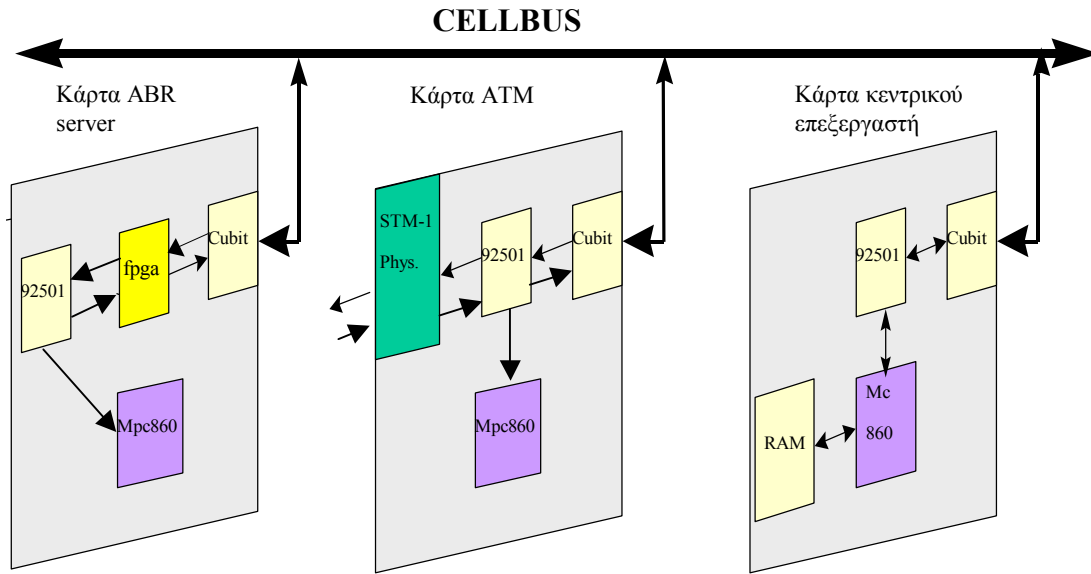
**Εικόνα 6 Μορφή επικεφαλίδας στο δίαυλο CellBus**

Τα δεδομένα τα οποία μεταδίδονται μέσω του διαύλου CellBus μπορεί να αποστέλλονται προς ένα, πολλούς ή όλες τις άλλες μονάδες. Σε αυτές τις μεταδόσεις χρησιμοποιούνται οι Single Address Data, Multicast Address Data και Broadcast Address Data επικεφαλίδας αντίστοιχα. Τα δεδομένα αυτού του τύπου παραδίδονται από το ολοκληρωμένο Cubit μέσω διεπαφής UTOPIA Level 1 [1] σε ολοκληρωμένα που συνδέονται με το Cubit μέσω της διεπαφής αυτής. Τα δεδομένα που μεταδίδονται με επικεφαλίδες Single Address Control, Multicast Address Control και Broadcast Address Control μεταδίδονται αντίστοιχα προς μία ή περισσότερες μονάδες αλλά κατά την λήψη τους εξάγονται από την ροή των άλλων δεδομένων και παραδίδονται σε μέσω διεπαφής μικροεπεξεργαστή (microprocessor interface).

Τα πεδία της επικεφαλίδας CellBus Routing Header περιέχουν πληροφορία για την φυσική διεπαφή που θα αποσταλεί το cell μέσω UTOPIA (πεδίο R), για την επιλογή της ουράς που θα αποθηκευτεί το cell (πεδίο Q), την διεύθυνση της μονάδας προορισμού στο δίαυλο (πεδίο A) και για την ομάδα πολλαπλής μετάδοσης (multicast group) προς την οποία αποστέλλεται το cell. Η επικεφαλίδα CellBus Routing Header περιέχει έλεγχο λαθών με την χρήση του πεδίου H, το οποίο περιέχει το CRC (Cyclic Redundancy Check) της επικεφαλίδας.

### 1.5 Οργάνωση και Λειτουργικότητα του συστήματος

Στην Εικόνα 7 φαίνεται η δομή των υποσυστημάτων του Κόμβου Πρόσβασης, η τοποθέτηση τους καθώς και ροή των cells μέσα στο σύστημα.

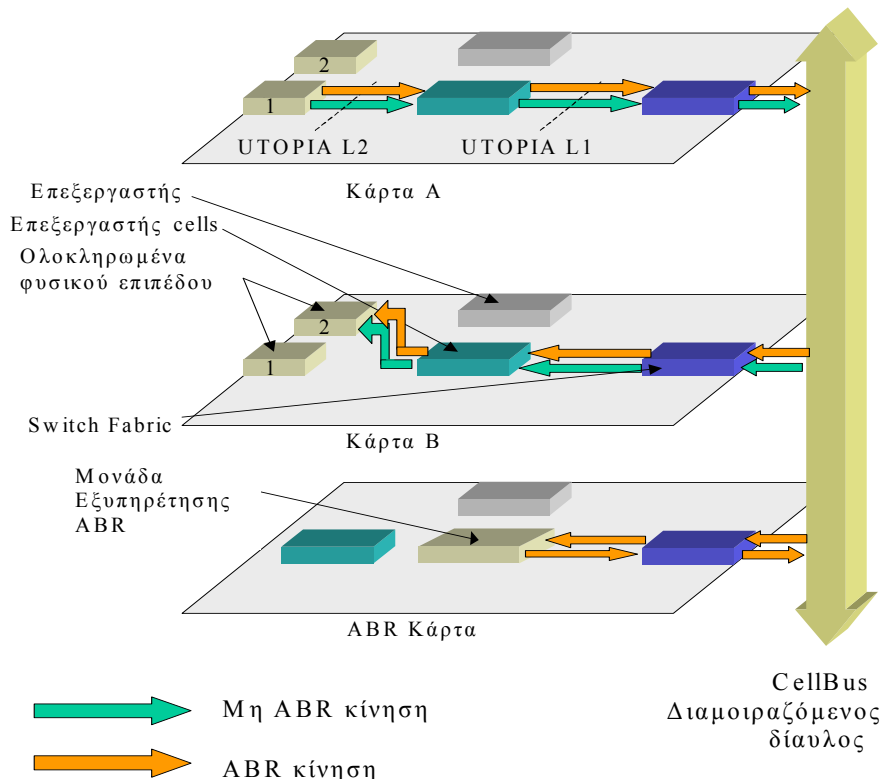


**Εικόνα 7** Δομή και τοποθέτηση των υποσυστημάτων του Κόμβου Πρόσβασης

Η λειτουργία του Κόμβου Πρόσβασης μέσω των καρτών που τον δομούν και των υποσυστημάτων, για κάθε επίπεδο λειτουργίας, περιγράφεται στις επόμενες παραγράφους.

### 1.5.1 Λειτουργικότητα Επιπέδου Χρήστη

Στην Εικόνα 8 φαίνονται οι ροές δεδομένων στο Επίπεδο Χρήστη.



**Εικόνα 8 Ροές Δεδομένων Επιπέδου Χρήστη**

Το υποσύστημα Φυσικού Επιπέδου των καρτών ΑΤΜ πραγματοποιούν την εξαγωγή των ΑΤΜ cells δεδομένων από την εισερχόμενη STM-1 ροή.

Τα cells στην συνέχεια περνούν μέσα από το *υποσύστημα επεξεργασίας κελιών*, όπου γίνεται η μετάφραση των κεφαλίδων τους. Πιο συγκεκριμένα, τα πεδία VPI/VCI μαζί με τον αριθμό του εισερχόμενου link χρησιμοποιούνται για να προσδιοριστεί ένα εσωτερικό προσδιοριστικό για την κάθε σύνδεση το οποίο ονομάζεται *Internal Connection Identifier (ICI)*[7]. Αυτό το προσδιοριστικό τοποθετείται στην θέση των πεδίων VP/VC και είναι μικρότερου μήκους (16 bit). Η μικρή και ευέλικτη δομή αυτού του πεδίου διευκολύνει σε μεγάλο βαθμό τον χειρισμό των cells στο εσωτερικό του Κόμβου Πρόσβασης και χρησιμοποιείται από τα διάφορα υποσυστήματα του ως αναγνωριστικό της σύνδεσης στην οποία ανήκει το κάθε cell. Μαζί με το ICI τοποθετείται στο εισερχόμενο cell και μία επιπλέον κεφαλίδα. Η επιπλέον κεφαλίδα αποτελείται από δύο πεδία. Το πρώτο πεδίο ονομάζεται *CellBus Routing header*, και χρησιμοποιείται από τα *στοιχεία μεταγωγής (CUBIT)* των καρτών και τον *κοινό διάυλο κελιών (CellBus)* ώστε να προσδιοριστεί η κάρτα προορισμού (βλ παρ. 1.4.3). Επίσης από το ίδιο πεδίο καθορίζεται και η προτεραιότητα των cells πάνω στον *κοινό διάυλο*. Το δεύτερο πεδίο ονομάζεται *CellBus Tandem header* και χρησιμοποιείται για cells που ανήκουν σε κίνηση τύπου ΑΒΡ. Σε αυτήν την περίπτωση το πεδίο *CellBus Routing header* περιέχει την διεύθυνση του *στοιχείου μεταγωγής* της κάρτας ΑΒΡ και το πεδίο *CellBus Tandem header* περιέχει την διεύθυνση του *στοιχείου μεταγωγής* της κάρτας εξόδου της κίνησης.

Όλη αυτή η πληροφορία δρομολόγησης αποθηκεύεται στην εξωτερική μνήμη (RAM) του *επεξεργαστή κελιών*, για κάθε ενεργή σύνδεση. Τα cells τοποθετούνται σε ουρές μέσα στο *στοιχείο μεταγωγής* (CUBIT), ανάλογα με την προτεραιότητα τους και το *στοιχείο μεταγωγής* με τη σειρά του πραγματοποιεί μία αίτηση για τον *κοινό δίαυλο* (CellBus). Ένα cell μπορεί να μεταφερθεί πάνω από το CellBus όταν το αιτών CUBIT δέχεται μία επιβεβαίωση (grant) από το CellBus. Τα cells γίνονται δεκτά από το CUBIT προορισμού, το οποίο προσδιορίζεται από τον CellBus header. Αυτή η κάρτα προορισμού μπορεί να είναι είτε ο ABR server(κίνηση ABR), είτε μία VDSL κάρτα ή τέλος μία άλλη κάρτα ATM (οποιοσδήποτε τύπος κίνησης εκτός ABR).

Στην περίπτωση του ABR server τα cells επεξεργάζονται και αποθηκεύονται σε ουρές και στη συνέχεια αποστέλονται ως έχουν στην κάρτα εξόδου. (Η λειτουργία αυτής της κάρτας περιγράφεται αναλυτικά στην επόμενη ενότητα.). Σε αυτή την περίπτωση η πορεία των cells από την είσοδο στην έξοδο γίνεται σε δύο βήματα (κάρτα εισόδου-ABR server-κάρτα εξόδου). Στο πρώτο βήμα χρησιμοποιείται το πρώτο πεδίο της κεφαλίδας (Routing Header) ως πληροφορία δρομολόγησης, ενώ στο δεύτερο βήμα χρησιμοποιείται ως κύρια πληροφορία δρομολόγησης το δεύτερο πεδίο (Tandem Header).

Στην περίπτωση μη ABR κίνησης τα cells γίνονται δεκτά απευθείας στην κάρτα εξόδου.

Σε όλες τις περιπτώσεις όταν τα cells φτάσουν στην κάρτα εξόδου περνούν μέσα από τον *επεξεργαστή κελιών*, όπου αφαιρείται η επιπλέον κεφαλίδα (Routing ή Tandem Header). Στη συνέχεια πραγματοποιείται το δεύτερο στάδιο της μετάφρασης διευθύνσεων όπου το ICI που περιέχουν τα cells χρησιμοποιείται για να καθοριστεί η τιμή του πεδίου VP/VC εξόδου η οποία τοποθετείται στα cells πρώτου αυτά εξαχθούν από την κάρτα.

Τέλος, το στοιχείο του φυσικού επιπέδου της κάρτας εξόδου εισάγει τα ATM cells δεδομένων στην ροή εξόδου.

### 1.5.2 Λειτουργικότητα Επιπέδου Ελέγχου

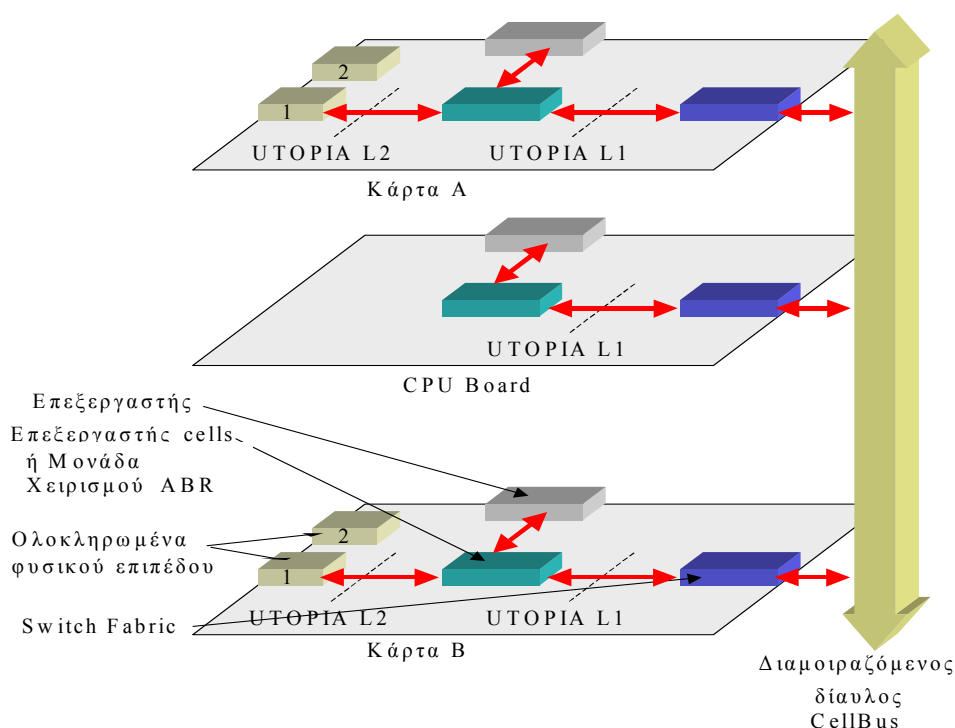
Οι ροές ελέγχου στο εσωτερικό του συστήματος AN-Switch περιλαμβάνουν το επίπεδο ελέγχου (control plane) του ATM, καθώς και την επικοινωνία ανάμεσα στις κάρτες του συστήματος (intercard communication) για μεταφορά πληροφοριών διαχείρισης (management plane) και επιτήρησης του συστήματος.

Στην Εικόνα 9 απεικονίζονται οι ροές επιπέδου ελέγχου κατά την διάρκεια εγκατάστασης μιας συνδέσεως. Στην φυσική διεπαφή 1 της κάρτας A λαμβάνεται ένα μήνυμα SETUP. Το μήνυμα έρχεται σε μορφή cells με VPI=0 και VCI=5. Τα ολοκληρωμένα της φυσικής διεπαφής προωθούν τα cells στο cell processor ο οποίος έχει προγραμματιστεί να εξάγει τα cell με τον συνδυασμό VPI=0 και VCI=5 και να τα προωθεί στο επεξεργαστή. Ο επεξεργαστής ειδοποιείται για την παραλαβή δεδομένων με χρήση σήματος διακοπής (interrupt) και διαβάσει το cell από το cell processor. Μετά την παραλαβή του τελευταίου cell του πακέτου, που διακρίνεται από το bit PTI=1 της ATM επικεφαλίδας, το λογισμικό (driver) αναλαμβάνει την δημιουργία

του πακέτου σύμφωνα με το AAL5 [6]. Το πακέτο παραδίδεται στο επίπεδο SAAL και στην συνέχεια στην σηματοδοσία της κάρτας A. Η σηματοδοσία αναγνωρίζει ότι πρόκειται για SETUP μήνυμα και μεταδίδει ένα SETUP \_indication προς τον έλεγχο αποδοχής κλήσεων που υλοποιείται στην CPU board του συστήματος. Για την μετάδοση του πακέτου το Access Signaling παραδίδει το SETUP \_indication στην οντότητα εσωτερικής επικοινωνίας η οποία παραδίδει το μήνυμα στον driver, ο οποίος το εισάγει σε μορφή cells στον επεξεργαστή cell με την ένδειξη ότι τα cells έχουν προορισμό την κεντρική κάρτα του συστήματος (CPU Board). Ο cell processor μεταδίδει μέσω της διεπαφής UTOPIA L1 τα cells προς το Cubit μαζί με την κατάλληλη επικεφαλίδα ώστε να μεταδοθούν στην CPU Board (τύπος Single Address Data στην Εικόνα 6).

Στην κεντρική κάρτα του συστήματος (CPU Board) τα ATM cells λαμβάνονται από το Cubit, αφαιρείται ο CellBus Routing Header, και μεταδίδονται στον cell processor, ο οποίος έχει προγραμματιστεί να τα παραδίδει στον επεξεργαστή. Ο επεξεργαστής λαμβάνει μέσω του driver το μήνυμα SETUP\_indication και η οντότητα ελέγχου αποδοχής κλήσεων δημιουργεί δύο μηνύματα. Το πρώτο είναι PROCEEDING\_request και αποστέλλεται προς την κάρτα A, και το δεύτερο είναι SETUP\_request και μεταδίδεται προς την κάρτα B στην οποία είναι συνδεδεμένος (ή μέσω της οποίας μπορεί να βρεθεί) ο χρήστη με την ATM address που υπήρχε στο SETUP που ελήφθη στην κάρτα A. Τα μηνύματα παραδίδονται στον driver με τα κατάλληλα αναγνωριστικά ώστε να δρομολογηθούν προς τις κάρτες A και B.

Τα cells του μηνύματος PROCEEDING\_request λαμβάνονται από το Cubit της κάρτας A και προωθούνται στο cell processor ο οποίος τα εξάγει και τα παραδίδει στον κεντρικό επεξεργαστή της κάρτας A. Το PROCEEDING\_request προκαλεί την δημιουργία ενός μηνύματος PROCEEDING από το Access Signaling το οποίο μεταδίδεται, μέσω του cell processor προς την φυσική διεπαφή στην οποία είχε παραληφθεί το αρχικό μήνυμα SETUP.



Εικόνα 9 Ροές ελέγχου

Τα cells του μηνύματος `SETUP_request` λαμβάνονται από το Cubit της κάρτας B και προωθούνται στο cell processor ο οποίος τα εξάγει και τα παραδίδει στον κεντρικό επεξεργαστή της κάρτας B. Το `SETUP_request` προκαλεί την δημιουργία ενός μηνύματος `SETUP` από το Access Signaling το οποίο μεταδίδεται, μέσω του cell processor προς την φυσική διεπαφή

Με το τέλος της διαδικασίας συνδέσεως η CPU Board αποστέλλει μηνύματα διαχείρισης τα οποία ειδοποιούν τις κάρτες A και B για τις παραμέτρους των ροών δεδομένων που πρέπει να αρχικοποιηθούν σε επίπεδο υλικού για την μεταφορά δεδομένων επιπέδου χρήστη (user plane). Η πληροφορία διαχείρισης χρησιμοποιεί τα αποτελέσματα της σηματοδοσίας όπως τα συμφωνηθέντα VPI/VCI, το εύρος ζώνης που θα αποδοθεί, αλλά περιέχει και πληροφορίες που σχετίζονται με την υλοποίηση του συστήματος, όπως το TAG και ο CellBus Routing Header που θα χρησιμοποιηθούν από τα ολοκληρωμένα του συστήματος για την μεταφορά δεδομένων του επιπέδου χρήστη (user plane).

### 1.5.3 Λειτουργίες Επιπέδου Διαχείρισης

Η Διαχείριση του Κόμβου Πρόσβασης χωρίζεται σε δύο μέρη, την Τοπική Διαχείριση και την Κεντρική Διαχείριση. Η Τοπική Διαχείριση πραγματοποιείται από τον επεξεργαστή που βρίσκεται σε κάθε κάρτα ενώ η Κεντρική Διαχείριση πραγματοποιείται από τον Κεντρικό Επεξεργαστή μέσω της επικοινωνίας του με τις επιμέρους κάρτες με την χρήση ειδικού πρωτοκόλλου.

Οι βασικές αρμοδιότητες της Κεντρικής Διαχείρισης είναι οι εξής:

- Αρχικοποίηση Συστήματος. Η αρχικοποίηση του συστήματος ξεκινά με την αρχικοποίηση της κάρτας Κεντρικού Επεξεργαστή, η οποία αποτελείται από την εκκίνηση (booting) του επεξεργαστή, το φόρτωμα (downloading) των προγραμμάτων και τον προγραμματισμό και τον έλεγχο των περιφερειακών υποσυστημάτων. Στη συνέχεια ο Κεντρικός Επεξεργαστής πληροφορείται για τις κάρτες που είναι παρούσες στο σύστημα και ολοκληρώνει την διαδικασία προγραμματισμού τους παρατηρώντας (polling) όλες τις CUBIT διευθύνσεις.
- Έλεγχος Καρτών. Ο Κεντρικός Επεξεργαστής πραγματοποιεί περιοδικά διαδικασίες ανίχνευσης των καρτών του συστήματος, με σκοπό να ελέγξει την κατάσταση στην οποία βρίσκονται ή ακόμη και την παρουσία τους στο σύστημα (σε περίπτωση που μία κάρτα αφαιρεθεί από το σύστημα). Οι νεοεισερχόμενες κάρτες στο σύστημα, ρυθμίζονται από τον Κεντρικό Επεξεργαστή, ενώ ταυτόχρονα απελευθερώνονται δεσμευμένοι πόροι όταν μία κάρτα εξάγεται από το σύστημα. Ο Κεντρικός Επεξεργαστής πληροφορεί τους τοπικούς επεξεργαστές των επιμέρους καρτών για την εγκατάσταση νέων συνδέσεων και τους παρέχει τις σχετικές παραμέτρους για αυτές τις συνδέσεις.. Τέλος ο Κεντρικός Επεξεργαστής δίνει εντολές στους τοπικούς επεξεργαστές και ανακτά στατιστικά δεδομένα ή άλλες πληροφορίες από αυτούς.
- Διαχείριση Σφαλμάτων. Πιθανά σφάλματα που ανιχνεύονται ή αναφέρονται, είτε διορθώνονται είτε απομονώνονται από τον Κεντρικό Επεξεργαστή.
- Εξωτερική επικοινωνία. Ο Κεντρικός Επεξεργαστής υποστηρίζει τα πρωτόκολλα για την εξωτερική επικοινωνία του συστήματος.

Οι βασικές αρμοδιότητες της Τοπικής Διαχείρισης είναι οι εξής:

- Αρχικοποίηση των καρτών.
- Ενημέρωση των υποσυστημάτων των καρτών για νέες συνδέσεις.
- Έλεγχος των υποσυστημάτων των καρτών.
- Διαχείριση Σφαλμάτων.
- Επικοινωνία με τον Κεντρικό Επεξεργαστή.

Η απόδοση της Διαχείρισης του Συστήματος εξαρτάται από τις δυνατότητες του πρωτοκόλλου επικοινωνίας μεταξύ του Κεντρικού Επεξεργαστή και των επεξεργαστών που βρίσκονται στις κάρτες του συστήματος. Να σημειωθεί ότι το πρωτόκολλο είναι υπεύθυνο για την μεταφορά πληροφορίας υψηλού επιπέδου, όπως πληροφορία για την κατάσταση των καρτών και εντολές CAC καθώς επίσης και πληροφορίας χαμηλού επιπέδου όπως περιεχόμενα της εσωτερικής μνήμης των καρτών. Το πρωτόκολλο χρησιμοποιεί ATM cells με αχρησιμοποίητα VPI/VCI ( VPI = 0, VCI = 1). Γενικά, η φιλοσοφία της σχεδίασης του συστήματος έχει ως στόχο την υλοποίηση ενός "μP Bus Protocol" πάνω από τις συνδέσεις ATM χρησιμοποιώντας τον κοινό δίαυλο κελιών (CellBus) (βλ. Παρ. 1.6).



### 1.5.4 Δρομολόγηση δεδομένων στο εσωτερικό του AN-Switch

Η δρομολόγηση των δεδομένων τα οποία μεταφέρονται με την μορφή cells διαιρείται στην δρομολόγηση σε επίπεδο κάρτας και σε δρομολόγηση σε επίπεδο συστήματος. Σε επίπεδο κάρτας τα cell μπορούν να μεταφερθούν από τις φυσικές διεπαφές προς το διαμοιραζόμενο δίαυλο CellBus μέσω του cell processor κατά την είσοδο τους στο σύστημα, ενώ ακολουθούν την αντίστροφη φορά δηλαδή από τον δίαυλο CellBus προς τις φυσικές διεπαφές κατά την έξοδο των δεδομένων από το AN-Switch. Τα ATM cells τα οποία ακολουθούν αυτές τις διαδρομές είναι δεδομένα του επιπέδου χρήστη (user plane). Τα cells των επιπέδων ελέγχου (control plane) και διαχείρισης (management plane) που λαμβάνονται από φυσική διεπαφή ή των διαμοιραζόμενο δίαυλο CellBus δρομολογούνται μέσω του cell processor στον επεξεργαστή της κάρτας. Σε επίπεδο συστήματος τα δεδομένα δρομολογούνται με βάση τον CellBus Routing Header, ο οποίος έχει εισαχθεί από το cell processor της κάρτας που μετέδωσε το cell. Η παραλαβή του cell γίνεται από μια ή περισσότερες κάρτες του συστήματος.

## 1.6 Εσωτερική επικοινωνία του AN-Switch

Ο ATM μεταγωγέας AN-Switch αποτελείται από ένα σύνολο καρτών (μέχρι 16), μεταξύ των οποίων γίνεται ανταλλαγή πληροφοριών με σκοπό την μεταγωγή δεδομένων αλλά και την ανταλλαγή μηνυμάτων για το συντονισμό και την διαχείριση του συστήματος. Διακρίνουμε δυο είδη εσωτερικών ροών δεδομένων που διακινούνται μέσω του διαμοιραζόμενου διαύλου CellBus.

Το πρώτο είδος εσωτερικής ροής αφορά τα δεδομένα ATM συνδέσεων τα οποία μεταγάζονται μέσω του CellBus. Τα δεδομένα αυτά προέρχονται από τις φυσικές διεπαφές, και περιέχουν πληροφορία που ανταλλάσσονται μεταξύ των τελικών χρηστών του ATM δικτύου. Η κίνηση τους στο εσωτερικό του μεταγωγέα γίνεται χωρίς έλεγχο από τις κεντρικές υπολογιστικές μονάδες (CPUs).

Το δεύτερο είδος εσωτερικής ροής αφορά πληροφορία η οποία ανταλλάσσεται μεταξύ των πρωτοκόλλων, με σκοπό τον συντονισμό και διαχείριση του συστήματος. Τέτοιου είδους επικοινωνία έχουν τα επίπεδα σηματοδοσίας τα οποία είναι καταναμημένα σε διαφορετικές κάρτες του συστήματος, καθώς και η τοπική διαχείριση.

### 1.6.1 Είδη μηνυμάτων εσωτερικής επικοινωνίας

Τα εσωτερικά μηνύματα που ανταλλάσσονται αφορούν δύο από τα επίπεδα του ATM: Το επίπεδο ελέγχου αποδοχής κλήσεων (CAC) και το επίπεδο διαχείρισης (management). Τα δυο αυτά επίπεδα δεν προδιαγράφονται από τις προδιαγραφές των διεθνών οργανισμών τυποποίησης και η υλοποίησή τους είναι άμεσα συνδεδεμένη με την εκάστοτε σχεδίαση.

Τα μηνύματα που ανταλλάσσονται ανάμεσα στις τοπικές σηματοδοσίες και στο κεντρικό έλεγχο αποδοχής κλήσεων, αναφέρονται σε σηματοδικά μηνύματα τα οποία είτε έχουν έρθει από φυσική διεπαφή είτε πρέπει να μεταδοθούν προς αυτήν. Ο

έλεγχος αποδοχής κλήσεων συντονίζει τις οντότητες σηματοδοσίας των καρτών του συστήματος.

Τα μηνύματα διαχείρισης αφορούν την κατάσταση του συστήματος, την διευθυνσιοδότηση των καρτών, και την εγκατάσταση/απεγκατάσταση των συνδέσεων σε επίπεδο υλικού.

Τα μέγεθος των μηνυμάτων είναι μεταβλητό και εξαρτάται από δυο παράγοντες: από την προέλευση του (από το access signaling ή το management) και από τον τύπο του μηνύματος. Για παράδειγμα ένα μήνυμα που ειδοποιεί τον έλεγχο αποδοχής κλήσης για μια εισερχόμενη κλήση μεταφέρει περισσότερο όγκο πληροφορίας από το αντίστοιχο που αναφέρεται στην απεγκατάστασή της.

Η συχνότητα μετάδοσης των εσωτερικών μηνυμάτων δεν είναι σταθερή, αλλά εξαρτάται άμεσα από το πλήθος των μηνυμάτων που εισέρχονται στο AN-Switch από τις φυσικές διεπαφές. Ο όγκος των μηνυμάτων είναι της τάξεως των εκατοντάδων bytes και συνδυασμό με την μικρή συχνότητα τους απαιτούν μικρό εύρος ζώνης για την μετάδοσή τους. Επίσης η εξυπηρέτηση των μηνυμάτων δεν έχει αυστηρούς χρονικούς περιορισμούς λόγω των χρόνων ανταπόκρισης που περιγράφονται από τα πρωτόκολλα σηματοδοσίας και είναι της τάξεως των δεκάδων δευτερολέπτων. Ακόμα οι λειτουργίες διαχείρισης θεωρούνται πάντα λειτουργίες μη πραγματικού χρόνου (non real time). Ωστόσο η αποδοτική διαχείριση και μεταφορά των εσωτερικών μηνυμάτων επιδρά άμεσα στην απόδοση του συστήματος σε παραμέτρους όπως ο αριθμός των κλήσεων ανά δευτερόλεπτο που μπορούν να εξυπηρετηθούν και ο χρόνος απεγκατάστασης συνδέσεων. Η αποδοτική μεταφορά και επεξεργασία των μηνυμάτων της εσωτερικής επικοινωνίας είναι επίσης σημαντική για την ευστάθεια του συστήματος, γιατί αποφεύγονται καταστάσεις που δημιουργούν προβλήματα λειτουργίας όπως υπερχειλίσεις ενταμιευτών που υλοποιούνται είτε σε υλικό είτε σε λογισμικό.

### 1.6.2 Φυσικό επίπεδο μεταφοράς μηνυμάτων

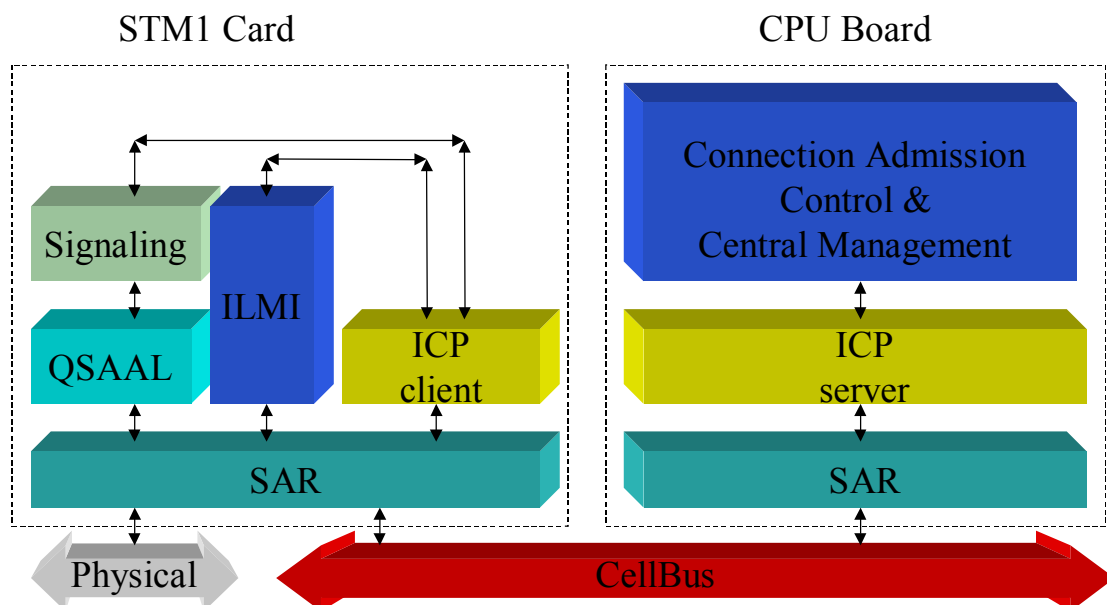
Η κατανεμημένη σχεδίαση και υλοποίηση των επιπέδων του ATM στο μεταγωγέα, προϋποθέτει την λογική σύνδεση των επιπέδων όπως έχει ήδη αναφερθεί, όσο και την ύπαρξη ενός φυσικού μέσου μεταφοράς. Μέσω αυτού θα γίνεται η μεταφορά των μηνυμάτων της εσωτερικής επικοινωνίας.

Το φυσικό μέσο μεταφοράς πρέπει να διασυνδέει όλες τις επεξεργαστικές μονάδες του AN-Switch (CPUs), οι οποίες βρίσκονται στις κάρτες του συστήματος. Η εκπομπή πρέπει να είναι δυνατή με απόφαση της τοπικής επεξεργαστικής μονάδας, όπως κατά την λήψη ενός μηνύματος σηματοδοσίας από μια διεπαφή φυσικού επιπέδου. Σε αυτήν την περίπτωση ο τοπικός επεξεργαστής που παρέλαβε το μήνυμα και μετά την ανάλογη επεξεργασία, θα ζητήσει την εκπομπή μηνύματος εσωτερικής επικοινωνίας προς το έλεγχο αποδοχής κλήσεων. Στην αντίστροφη φορά κατά την παραλαβή μηνύματος εσωτερικής επικοινωνίας ο επεξεργαστής θα πρέπει να είναι σε θέση να διακόψει την εκτέλεση του προγράμματος που εκτελεί εκείνη την στιγμή και να εξυπηρετήσει τον ληφθέν μήνυμα.

Το εύρος ζώνης που απαιτείται από το φυσικό μέσο μεταφοράς είναι της τάξεως των δεκάδων Kbps όπως αναλύθηκε στην προηγούμενη παράγραφο. Η χρήση ενός φυσικού μέσου μεταφοράς κρίθηκε μη ικανοποιητική σε σχέση με το κόστος σχεδίασης και υλοποίησης της, δεδομένου ότι όλες οι κάρτες του συστήματος διασυνδέονται μέσω του διαμοιραζόμενου δίαυλου CellBus. Ο δίαυλος έχει ικανότητα μεταγωγής της τάξης του ενός Gbps, και συνεπώς είναι εφικτή η χρήση του για μεταφορά πληροφοριών της τάξεως των Kbps. Επίσης σε περίπτωση που υπάρχει καταγισμός αιτήσεων εγκατάστασης συνδέσεως, ο δίαυλος μπορεί να διακινήσει πληροφορία της τάξεως των Mbps. Λαμβάνοντα υπόψη ότι η λειτουργία του CellBus στο σύστημα AN-Switch είναι δεδομένη, η χρήση του και για την μεταφορά των πληροφοριών εσωτερικής επικοινωνίας έχει σαν αποτέλεσμα την βελτίωση της χρησιμοποίησης (utilization) του συστήματος.

### 1.6.3 Περιγραφή πρωτοκόλλου εσωτερικής επικοινωνίας

Το πρωτόκολλο εσωτερικής επικοινωνίας (Internal Communication Protocol, ICP) σχεδιάστηκε για τις ανάγκες της εσωτερικής επικοινωνίας του AN-Switch. Σκοπός της εσωτερικής επικοινωνίας είναι η διασύνδεση των οντοτήτων σηματοδοσίας και διαχείρισης με τον κεντρικό έλεγχο αποδοχής κλήσεων και την κεντρική διαχείριση, καθώς και η παροχή υπηρεσιών διαχείρισης χαμηλού επιπέδου. Η επικοινωνία αυτή γίνεται μεταξύ κάθε κάρτας VDSL ή STM1 ή ABR server με την κεντρική CPU Board. Η μορφή αυτή όσο και οι ανάγκες των υπηρεσιών που πρέπει να προσφέρει το ICP οδηγούν σε ένα client/server πρωτόκολλο, όπου κάθε κάρτα VDSL ή STM1 ή ABR server εκτελεί την client πλευρά και η κεντρική CPU κάρτα εκτελεί την master πλευρά. Η θέση του ICP στην αρχιτεκτονική του AN-Switch απεικονίζεται στην Εικόνα 10.

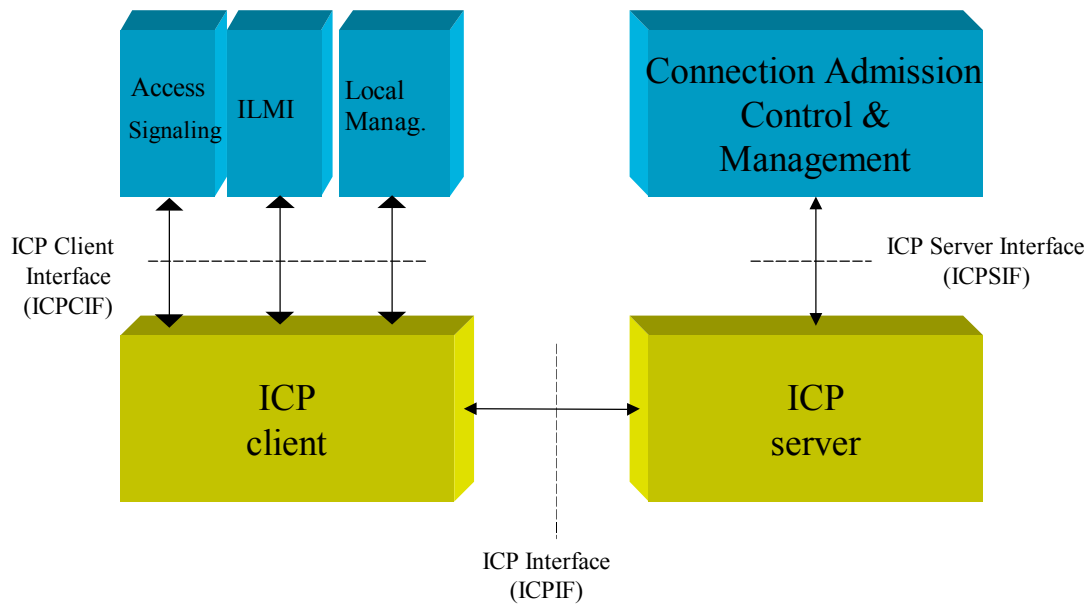


Εικόνα 10 Η θέση του ICP στην αρχιτεκτονική λογισμικού του AN-Switch

Όπως έχει αναφερθεί τα μηνύματα εσωτερικής επικοινωνίας είναι μεταβλητού μεγέθους. Τα μηνύματα εισάγονται/εξάγονται από την ροή των δεδομένων μέσω του cell processor χρησιμοποιώντας τον αντίστοιχο driver ο οποίος έχει δυνατότητα για κατάτμηση και επανένωση (segmentation and reassembly) AAL5 [6] πακέτων μεταβλητού μεγέθους. Το μήνυμα σε μορφή cells μεταγόνται στην κάρτα προορισμού όπου επανασχηματίζεται το AAL5 πακέτο και προωθείται στο επεξεργαστή της κάρτας.

Για την μεταγωγή των cells που μεταφέρουν μηνύματα εσωτερικής επικοινωνίας χρησιμοποιείται ένα ειδικό TAG του cell processor, το οποίο δεσμεύεται ως μη χρησιμοποιούμενο από τις ροές δεδομένων χρήστη που μεταγόνται από το AN-Switch.

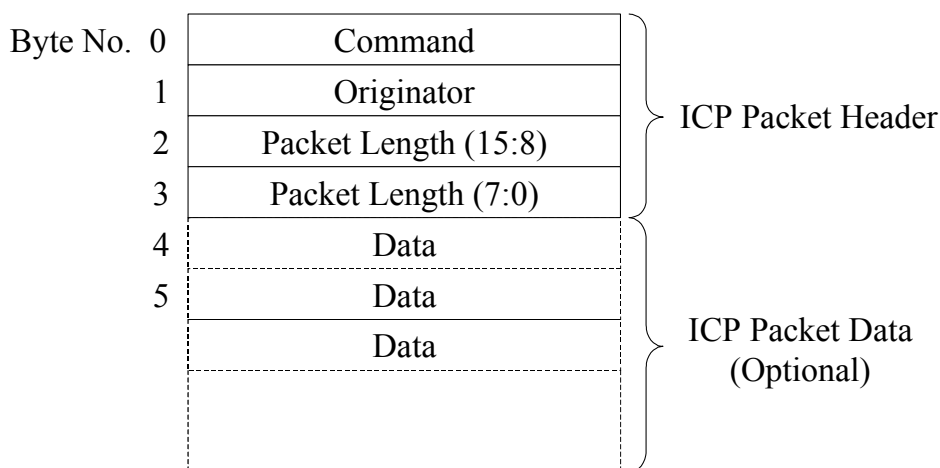
Διακρίνουμε δυο διεπαφές του ICP. Η πρώτη είναι αυτή που χρησιμοποιείται για την ανταλλαγή των μηνυμάτων ανάμεσα σε δυο ICP οντότητες. Η διεπαφή αυτή ονομάζεται ICP InterFace (ICPIF). Η δεύτερη είναι προς τις οντότητες λογισμικού υψηλότερου επιπέδου, που χρησιμοποιούν τις υπηρεσίες του ICP. Διακρίνουμε δύο διεπαφές η μια προς τις οντότητες υψηλότερου επιπέδου στην πλευρά του client που ονομάζεται ICP Client InterFace (ICPCIF) και μια προς τις οντότητες υψηλότερου επιπέδου στην πλευρά του server που ονομάζεται ICP Server InterFace (ICPSIF). Οι διεπαφές του ICP απεικονίζονται στην Εικόνα 11.



Εικόνα 11 Οι διεπαφές του ICP

### 1.6.3.1 Περιγραφή μηνυμάτων ομότιμων οντοτήτων

Το format των μηνυμάτων εσωτερικής επικοινωνίας που ανταλλάσσονται ανάμεσα στις ICP οντότητες, μέσω του ICP interface απεικονίζεται στην Εικόνα 12.



**Εικόνα 12 Η κωδικοποίηση των μηνυμάτων στο ICPIF**

Τα μηνύματα του ICP είναι μεταβλητού μεγέθους. Τα πρώτα 4 bytes είναι η επικεφαλίδα του ICP μηνύματος. Το πρώτο byte είναι η εντολή (command) προς το ICP για την αποκωδικοποίηση του μηνύματος (Εικόνα 12). Το δεύτερο byte περιέχει την διεύθυνση του αποστολέα του μηνύματος, η οποία ταυτίζεται με την διεύθυνση του Cubit της κάρτας στην οποία βρίσκεται. Το τρίτο και τέταρτο byte περιέχουν το συνολικό μέγεθος του μηνύματος (συμπεριλαμβανομένου του ICP header). Στην συνέχεια ακολουθούν οι πληροφορίες του μηνύματος η αποκωδικοποίηση των οποίων συναρτάται από πεδίο εντολής του μηνύματος.

Κωδικός (binary)	Κωδικός (hexadecimal)	Ονομασία μηνύματος
00000001	0x01	ICP_PING
00000010	0x02	ICP_PING_REPLY
00000110	0x06	ICP_OPEN_CONNECTION
00000111	0x07	ICP_CLOSE_CONNECTION
00001000	0x08	ICP_OPEN_DATA_CHAN
00001001	0x09	ICP_DEL_DATA_CHAN
10000000	0x80	ICP_FRWD_MANGMT
11000000	0xC0	ICP_FRWD_SIGNALING

**Πίνακας 2 Κωδικοί του Command Field στο ICPIF**

- **ICP\_PING (0x01):** Μεταδίδεται περιοδικά από το server ICP με σκοπό τον έλεγχο της ορθής λειτουργίας των καρτών του συστήματος και την εγγραφή τους ως ενεργές από την κεντρική διαχείριση. Η κάρτα που λαμβάνει ένα ICP\_PING

πρέπει να απαντήσει με ICP\_PING\_REPLY αλλιώς θα θεωρηθεί ως μη ενεργή από την κεντρική CPU board. Μέγεθος μηνύματος: 4 bytes.

- **ICP\_PING\_REPLY (0x02):** Το μήνυμα αυτό μεταδίδεται από client ICP σε απάντηση του ICP\_PING του server ICP. Περιλαμβάνει ένα byte πληροφορίας που αναφέρει αν είναι STM1 (τιμή 1) κάρτα ή E1 (τιμή 2). Μέγεθος μηνύματος: 5 bytes.
- **ICP\_OPEN\_CONNECTION (0x06):** Το μήνυμα αυτό μεταδίδεται από την κεντρική κάρτα προς client ICP της κάρτας ABR με σκοπό την αρχικοποίηση (εγκατάσταση) μιας ABR σύνδεσης για χειρισμό από την κάρτα ή την ενημέρωση για εγκατάσταση μη ABR σύνδεσης και των νέων παραμέτρων που αυτή διαμορφώνει κυρίως όσον αφορά τους διαθέσιμους πόρους του συστήματος. Η πληροφορίες που περιέχονται στο μήνυμα είναι (Εικόνα 13):

**TAGI (15:0):** Πεδίο 16 bits, που περιέχει την τιμή του εσωτερικού TAG, που θα τοποθετείται στην θέση της ATM επικεφαλίδας, στην μία κατεύθυνση (πλευρά εισόδου της κλήσης). Αυτό το πεδίο υπάρχει μόνο στην εγκατάσταση σύνδεσης ABR.

**TAGO (15:0):** Πεδίο 16 bits, που περιέχει την τιμή του εσωτερικού TAG, που θα τοποθετείται στην θέση της ATM επικεφαλίδας, στην μία κατεύθυνση (πλευρά εξόδου της κλήσης). Αυτό το πεδίο υπάρχει μόνο στην εγκατάσταση σύνδεσης ABR.

**PHYI (7:0):** Πεδίο 8 bits, που περιέχει τον αριθμό της φυσικής διεπαφής από την οποία λαμβάνονται (και αποστέλλονται στην ανάστροφη φορά) τα δεδομένα (πλευρά εισόδου της κλήσης).

**PHYO (7:0):** Πεδίο 8 bits, που περιέχει τον αριθμό της φυσικής διεπαφής από την οποία λαμβάνονται (και αποστέλλονται στην ανάστροφη φορά) τα δεδομένα (πλευρά εξόδου της κλήσης).

**Connection Type:** Πεδίο 8 bits που περιγράφει τον τύπο της νέας σύνδεσης (ABR, non-ABR) για την οποία πληροφορείται ο ABR server.

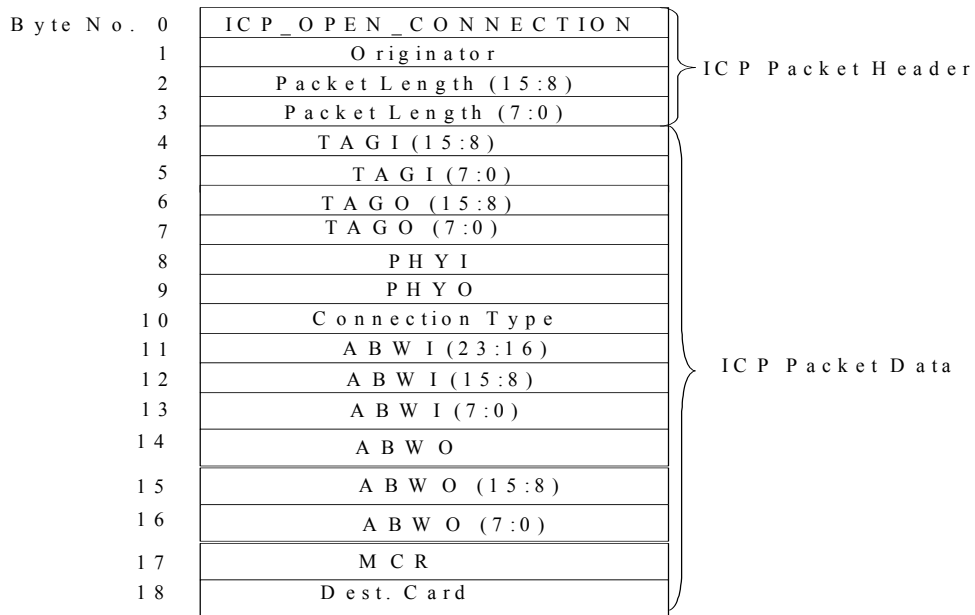
**ABWI(23:0):** Πεδίο 24 bits που περιέχει το διαθέσιμο ευρος ζώνης για τις ABR συνδέσεις στην φυσική διεπαφή PHYI

**ABWO(23:0):** Πεδίο 24 bits που περιέχει το διαθέσιμο ευρος ζώνης για τις ABR συνδέσεις στην φυσική διεπαφή PHYO

**Dest\_Card:** Πεδίο 8 bits που περιέχει την διεύθυνση του αποδέκτη των cells της συνδέσεως δεδομένων. Χρησιμοποιείται μόνο στις point to point συνδέσεις.

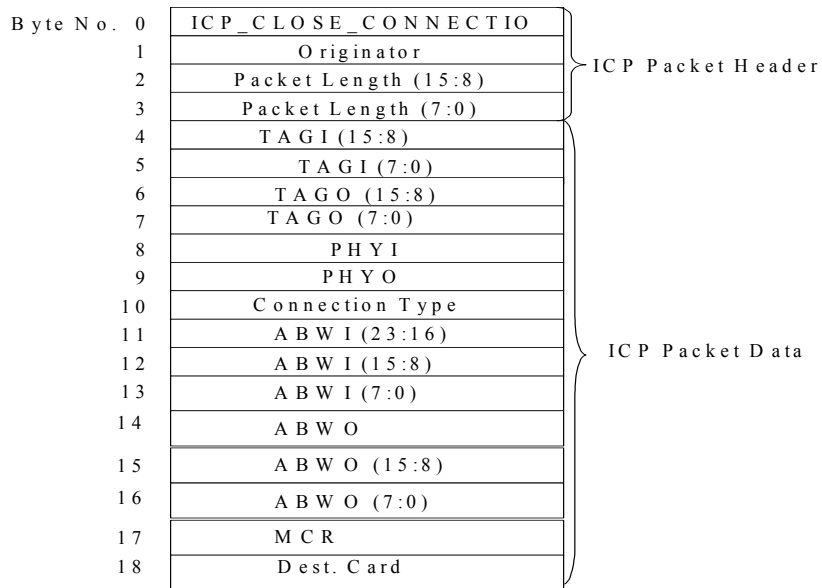
**MCR:** Πεδίο x bits που περιέχει το Minimum Cell Rate της σύνδεσης εφόσον αυτή είναι ABR. Αυτό το πεδίο υπάρχει μόνο στην εγκατάσταση σύνδεσης ABR.

Μέγεθος μηνύματος: 19 bytes (για ABR συνδέσεις), 14 bytes (για μη ABR συνδέσεις).



**Εικόνα 13 Η μορφή του μηνύματος ICP\_OPEN\_CONNECTION**

- **ICP\_CLOSE\_CONNECTION (0x07):** Το μήνυμα αυτό μεταδίδεται από την κεντρική κάρτα προς client ICP με σκοπό την απεγκατάσταση μιας σύνδεσης στον ABR server (Εικόνα 14). Μέγεθος μηνύματος: 19 bytes (για ABR συνδέσεις), 14 bytes (για μη ABR συνδέσεις).



**Εικόνα 14 Η μορφή του μηνύματος ICP\_CLOSE\_CONNECTION**

- **ICP\_OPEN\_DATA\_CHAN (0x08):** Το μήνυμα αυτό μεταδίδεται από την κεντρική κάρτα προς client ICP με σκοπό την αρχικοποίηση (εγκατάσταση) μιας ροής δεδομένων για μεταγωγή. Η πληροφορίες που περιέχονται στο μήνυμα είναι (Εικόνα 15):

VPIin (15:0): Πεδίο 16 bits, το οποίο περιέχει το VPI των cells που εισέρχονται από την φυσική διεπαφή της προς εγκατάστασης συνδέσεως.

VCIin (15:0): Πεδίο 16 bits, το οποίο περιέχει το VCI των cells που εισέρχονται από την φυσική διεπαφή της προς εγκατάστασης συνδέσεως.

PHYin (7:0): Πεδίο 8 bits, που περιέχει τον αριθμό της φυσικής διεπαφής από την οποία λαμβάνονται τα δεδομένα.

TAG (15:0): Πεδίο 16 bits, που περιέχει την τιμή του εσωτερικού TAG, που θα τοποθετείται στην θέση της ATM επικεφαλίδας.

VPIout (15:0): Πεδίο 16 bits, το οποίο περιέχει το VPI των cells που εξέρχονται από την φυσική διεπαφή της προς εγκατάστασης συνδέσεως.

VCIout (15:0): Πεδίο 16 bits, το οποίο περιέχει το VCI των cells που εξέρχονται από την φυσική διεπαφή της προς εγκατάστασης συνδέσεως.

PHYout (7:0): Πεδίο 8 bits, που περιέχει τον αριθμό της φυσικής διεπαφής από την οποία αποστέλονται τα δεδομένα.

MGROUP (7:0): Πεδίο 8 bits, που περιέχει την τιμή του multicast group που ανήκει η σύνδεση. Η τιμή 0 δεν είναι έγκυρη και σημαίνει ότι η προς εγκατάσταση σύνδεση είναι point to point.

MSource (7:0): Πεδίο 8 bits, που περιγράφει αν η κάρτα που έλαβε το μήνυμα είναι η πηγή της multicast συνδέσεως. Αν MSource=1 η κάρτα είναι πηγή της multicast συνδέσεως, αλλιώς αν MSource=0 είναι μέλος της multicast συνδέσεως. Η διάκριση για το αν μια κάρτα είναι η πηγή ή όχι μιας multicast συνδέσεως σχετίζεται με τις διαδικασίες προγραμματισμού που πρέπει να γίνουν στα ολοκληρωμένα της κάρτας. Αν η κάρτα είναι πηγή της multicast συνδέσεως τότε πρέπει να ενεργοποιήσει τον cell processor να λαμβάνει τα cells από την φυσική διεπαφή PHY με τιμές VPI/VCI να εισάγει το TAG, και επίσης να εισάγει την αντίστοιχη επικεφαλίδα (με βάση την τιμή του πεδίου MGROUP) στις πληροφορίες που προστίθενται από τον cell processor σε κάθε cell της συνδέσεως. Αν η κάρτα είναι μέλος μιας multicast συνδέσεως τότε πρέπει να προγραμματίσει το Cubit της να λαμβάνει τα cells με την επικεφαλίδα MGROUP και το cell processor να αποδέχεται τα cells με τιμή TAG, και να τα προωθεί προς τη φυσική διεπαφή PHY με τιμές VPI/VCI.

Dest\_Card: Πεδίο 8 bits που περιέχει την διεύθυνση του αποδέκτη των cells της συνδέσεως δεδομένων. Χρησιμοποιείται μόνο στις point to point συνδέσεις. Στις multipoint συνδέσεις (MGROUP≠0) το πεδίο υπάρχει στο μήνυμα αλλά δεν είναι έγκυρο. Η πληροφορία που υπάρχει σε αυτό το πεδίο χρησιμοποιείται για τον προγραμματισμό του cell processor.

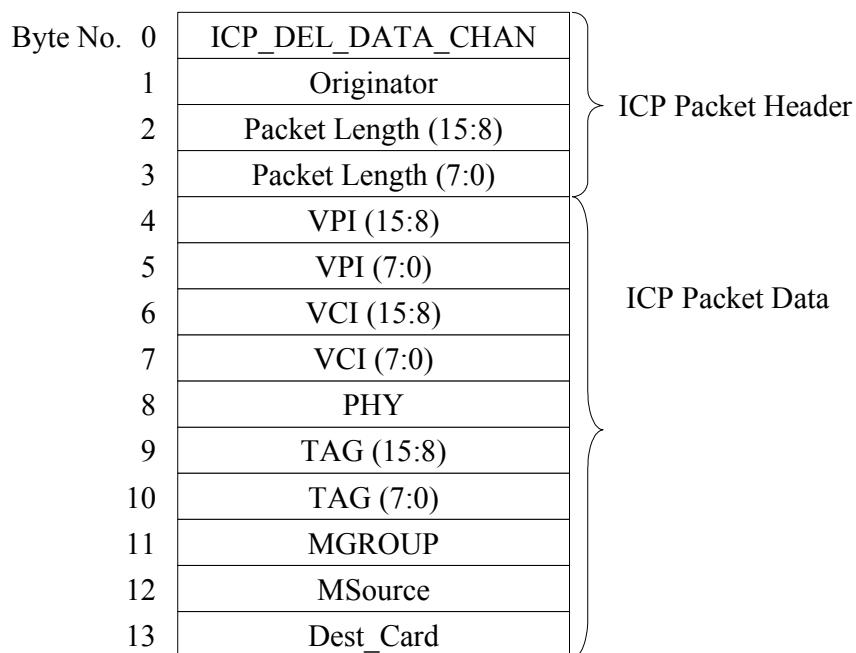
Μέγεθος μηνύματος: 14 bytes.



Byte No.	0	ICP_OPEN_DATA_CHAN	}	ICPIF Packet Header
	1	Originator		
	2	Packet Length (15:8)		
	3	Packet Length (7:0)	}	ICPIF Packet Data
	4	VPIin (15:8)		
	5	VPIin (7:0)		
	6	VCIin (15:8)		
	7	VCIin (7:0)		
	8	PHYin		
	9	TAG (15:8)		
	10	TAG (7:0)		
	11	VPIout (15:8)		
	12	VPIout (7:0)		
	13	VCIout (15:8)		
	14	VCIout (7:0)		
	15	PHYout		
	16	MGROUP		
	17	MSource		
	18	Dest_Card		

**Εικόνα 15 Η μορφή του μηνύματος ICP\_OPEN\_DATA\_CHAN**

- **ICP\_DEL\_DATA\_CHAN (0x09):** Το μήνυμα αυτό μεταδίδεται από την κεντρική κάρτα προς client ICP με σκοπό την διαγραφή (απεγκατάσταση) μιας ροής δεδομένων που μεταγίνονται. Η πληροφορίες που περιέχονται στο μήνυμα (Εικόνα 16) είναι οι ίδιες με αυτές του ICP\_OPEN\_DATA\_CHAN. Κατά την απεγκατάσταση αν MGROUP≠0 και MSource=0 τότε εκτός από την διαγραφή του TAG στην μνήμη του cell processor γίνεται και απεγκατάσταση του MGROUP από το Cubit ώστε να μην λαμβάνει δεδομένα από την multicast σύνδεση.

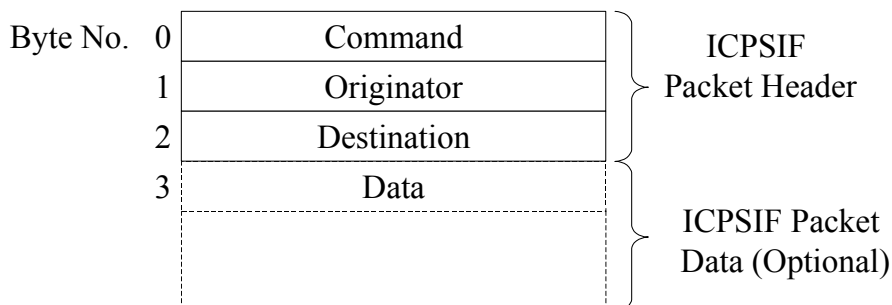


Εικόνα 16 Η μορφή το μηνύματος ICP\_DEL\_DATA\_CHAN

- **ICP\_FRWD\_MANGMT (0x80):** Το μήνυμα περιέχει στο πεδίο δεδομένων ένα πακέτο για την οντότητα διαχείρισης της κάρτας. Στην server η οντότητα διαχείρισης είναι η κεντρική διαχείριση του συστήματος, ενώ στην πλευρά του client είναι το ILMI.
- **ICP\_FRWD\_SIGNALING (0xC0):** Το μήνυμα περιέχει στο πεδίο δεδομένων ένα πακέτο για την οντότητα σηματοδοσίας της κάρτας. Στην server η οντότητα σηματοδοσίας είναι ο έλεγχος αποδοχής κλήσεων, ενώ στην πλευρά του client είναι το Access Signaling.

### 1.6.3.2 Περιγραφή μηνυμάτων προς άλλες οντότητες

Το format των μηνυμάτων που ανταλλάσσονται ανάμεσα στην ICP server οντότητα και στην κεντρική οντότητα διαχείρισης και αποδοχής κλήσεων, μέσω του ICPSIF interface απεικονίζεται στην Εικόνα 17.



Εικόνα 17 Η κωδικοποίηση των μηνυμάτων στο ICPSIF

Τα μηνύματα στο ICPSIF είναι μεταβλητού μεγέθους. Τα πρώτα 3 bytes είναι η επικεφαλίδα του ICP μηνύματος. Το πρώτο byte είναι η εντολή (command) για την

αποκωδικοποίηση του μηνύματος (Εικόνα 17). Το δεύτερο byte περιέχει την διεύθυνση του αποστολέα του μηνύματος, η οποία ταυτίζεται με την διεύθυνση του Cubit της κάρτας στην οποία βρίσκεται. Το τρίτο byte περιέχει την διεύθυνση προορισμού του μηνύματος. Όταν το server ICP παραδίδει μήνυμα στην κεντρική οντότητα διαχείρισης και αποδοχής κλήσεων το πεδίο Destination ισούται με τη διεύθυνση του Cubit της κεντρικής κάρτας, και συνεπώς δεν χρήζει επεξεργασίας. Στην αντίστροφη κατεύθυνση, δηλαδή όταν παραδίδεται ένα μήνυμα από την κεντρική οντότητα διαχείρισης και αποδοχής κλήσεων προς το server ICP, το πεδίο destination είναι αυτό που θα καθορίσει σε ποια κάρτα του συστήματος πρέπει να παραδοθεί το μήνυμα.

Κωδικός (binary)	Κωδικός (hexadecimal)	Ονομασία μηνύματος
00000001	0x01	ICP_NEW_CARD_FOUND
00000010	0x02	ICP_CARD_NOT_RESPOND
00001000	0x08	ICP_OPEN_DATA_CHAN
00001001	0x09	ICP_DEL_DATA_CHAN
10000000	0x80	ICP_FRWD_MANGMT
11000000	0xC0	ICP_FRWD_SIGNALING

**Πίνακας 3 Κωδικοί του Command field στο ICPSIF**

- **ICP\_NEW\_CARD\_FOUND (0x01):** Το μήνυμα αυτό μεταδίδεται από το server ICP προς την κεντρική διαχείριση όταν απαντήσει μια κάρτα του συστήματος, η οποία μέχρι εκείνη τη στιγμή ήταν ανενεργή, με ICP\_PING\_REPLY. Το μήνυμα περιέχει δύο byte πληροφορίας. Το πρώτο περιέχει την διεύθυνση της κάρτας στο CellBus, και το δεύτερο τον τύπο της κάρτας με τιμή 1 αν είναι STM1 ή με τιμή 2 αν είναι κάρτα E1. Μέγεθος μηνύματος: 5 bytes.
- **ICP\_CARD\_NOT\_RESPOND (0x02):** Το μήνυμα αυτό μεταδίδεται από το server ICP προς την κεντρική διαχείριση όταν δεν ανταποκρίνεται μια κάρτα του συστήματος, η οποία μέχρι εκείνη τη στιγμή ήταν ενεργή, σε μηνύματα ICP\_PING. Το μήνυμα περιέχει ένα byte πληροφορίας που περιέχει την διεύθυνση της κάρτας στο CellBus. Μέγεθος μηνύματος: 5 bytes.
- **ICP\_OPEN\_DATA\_CHAN (0x08):** Το μήνυμα αυτό μεταδίδεται από την κεντρική οντότητα διαχείρισης και αποδοχής κλήσεων κάρτα προς το server ICP με σκοπό την αρχικοποίηση (εγκατάσταση) μιας ροής δεδομένων για μεταγωγή. Η πληροφορίες που περιέχονται στο μήνυμα απεικονίζονται στην Εικόνα 18.

Byte No.	0	ICP_OPEN_DATA_CHAN	ICPSIF Packet Header
	1	Originator	
	2	Destination	
	3	VPIin (15:8)	
	4	VPIin (7:0)	
	5	VCIin (15:8)	
	6	VCIin (7:0)	
	7	PHYin	
	8	TAG (15:8)	
	9	TAG (7:0)	ICPSIF Packet Data
	10	VPIout (15:8)	
	11	VPIout (7:0)	
	12	VCIout (15:8)	
	13	VCIout (7:0)	
	14	PHYout	
	15	MGROUP	
	16	MSource	
	17	Dest_Card	

**Εικόνα 18** Μορφή του μηνύματος ICP\_OPEN\_DATA\_CHAN στο ICPSIF

Ο ICP server με την λήψη του μηνύματος αυτού πρέπει να δημιουργήσει ένα μήνυμα ICP\_OPEN\_DATA\_CHAN για το ICPIF.

- **ICP\_DEL\_DATA\_CHAN (0x09)**: Το μήνυμα αυτό μεταδίδεται από την κεντρική οντότητα διαχείρισης και αποδοχής κλήσεων κάρτα προς το server ICP με σκοπό την απεγκατάσταση μιας ροής δεδομένων για μεταγωγή. Η πληροφορίες που περιέχονται στο μήνυμα απεικονίζονται στην Εικόνα 19.

Byte No.	0	ICP_DEL_DATA_CHAN	ICPSIF Packet Header
	1	Originator	
	2	Destination	
	3	VPI (15:8)	ICPSIF Packet Data
	4	VPI (7:0)	
	5	VCI (15:8)	
	6	VCI (7:0)	
	7	PHY	
	8	TAG (15:8)	
	9	TAG (7:0)	
	10	MGROUP	
	11	MSource	
	12	Dest_Card	

**Εικόνα 19** Η μορφή του μηνύματος ICP\_DEL\_DATA\_CHAN στο ICPSIF

Ο ICP server με την λήψη του μηνύματος αυτού πρέπει να δημιουργήσει ένα μήνυμα ICP\_DEL\_DATA\_CHAN για το ICPIF.

- **ICP\_FRWD\_MANGMT (0x80)**: Το μήνυμα περιέχει στο πεδίο δεδομένων ένα πακέτο για την οντότητα διαχείρισης της destination κάρτας. Το server ICP πρέπει το μήνυμα αυτό να το ενθυλακώσει σε ένα μήνυμα ICP\_FRWD\_MANGMT για το ICPIF.
- **ICP\_FRWD\_SIGNALING (0xC0)**: Το μήνυμα περιέχει στο πεδίο δεδομένων ένα πακέτο για την οντότητα σηματοδότησης της destination κάρτας. Το server ICP πρέπει το μήνυμα αυτό να το ενθυλακώσει σε ένα μήνυμα ICP\_FRWD\_SIGNALING για το ICPIF.

## 2 Ο Εξυπηρετητής ABR κίνησης

Η ABR υπηρεσία των δικτύων ATM έρχεται να καλύψει τις ανάγκες των παραδοσιακών δικτύων LAN. Τα υπολογιστικά συστήματα σε ένα LAN θέλουν να στείλουν τα δεδομένα τους, τη στιγμή που αυτά γενιούνται και με ταχύτητα, αν αυτό είναι δυνατό, αυτή της γραμμής αλλά χωρίς συνωστισμό (congestion) που προκαλεί απώλεια cells. Αυτό ισχύει γιατί τα δεδομένα των Ηλεκτρονικών Υπολογιστών είναι ευαίσθητα στις απώλειες, μιας και οι αναμεταδώσεις μπορεί να μειώσουν δραστικά την απόδοση όλου του δικτύου.

Αντί λοιπόν να δεσμεύονται πόροι για την εκρηκτική κίνηση (bursty traffic) των LAN δικτύων, αυτή εξυπηρετείται από την υπηρεσία ABR που χρησιμοποιεί την διαθέσιμη κίνηση που δεν χρησιμοποιούν οι υπόλοιπες, ευαίσθητες σε χρονισμό, υπηρεσίες του ATM δικτύου. Προκειμένου όμως να δωθεί στους χρήστες η ικανότητα να στέλνουν ότι θέλουν όταν το θέλουν, με μόνη απαίτηση να μην υπάρχουν απώλειες, η εξυπηρέτηση της κίνησης ABR πρέπει να γίνεται από ένα μεταγωγέα που διαθέτει μεγάλη αποθηκευτική ικανότητα, ώστε να μπορούν να αποθηκευτούν για κάποιο χρονικό διάστημα τα cells που δεν μπορούν να μεταδοθούν λόγω μη διαθέσιμης κίνησης.

Γι' αυτό, στο μεταγωγέα που σχεδιάζουμε, χρησιμοποιούμε ένα κεντρικοποιημένο εξυπηρετητή κίνησης ABR, ώστε να μπορούμε να εκμεταλευτούμε ικανοποιητικά τους πόρους μνήμης με χαμηλό κόστος. Πιο συγκεκριμένα η κάρτα εξυπηρέτησης ABR κίνησης διαθέτει ικανότητες αποθήκευσης της συνολικής κίνησης ABR που δέχεται ο μεταγωγέας. Με αυτό τον τρόπο δεν παρενοχλείται η εξυπηρέτηση των άλλων ειδών κίνησης και η μνήμη της κάρτας χρησιμοποιείται αποκλειστικά για κίνηση ABR, που προωθείται όταν ο δίαυλος του μεταγωγέα είναι ελεύθερος (όταν δηλαδή δεν υπάρχει άλλου είδους κίνησης) απλοποιώντας έτσι την διαχείριση της κίνησης στις κάρτες γραμμής. Έτσι η κεντρικοποιημένη εξυπηρέτηση ABR μειώνει το κόστος μνήμης μιας και δεν τοποθετούμε μνήμη σε κάθε κάρτα γραμμής.

Αν και η κεντρικοποιημένη εξυπηρέτηση ABR δίνει μοναδικό σημείο αποτυχίας, το πρόβλημα μπορεί να λυθεί με την χρήση δεύτερης κάρτας που θα παρέχει ανοχή σε σφάλματα και μοίρασμα φόρτου.

### 2.1 Δομή και Λειτουργικότητα της κάρτας Εξυπηρετητή ABR

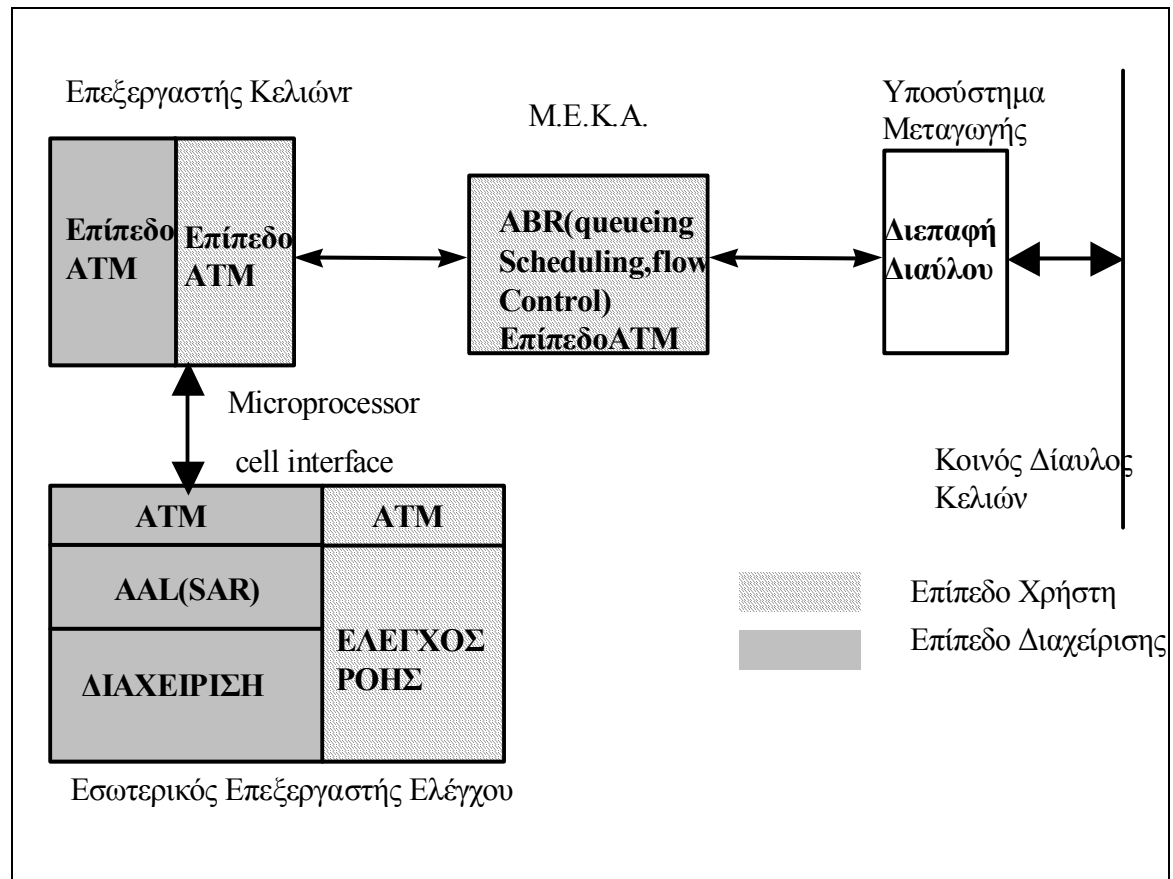
Η κάρτα Εξυπηρετητή ABR ως μέρος του συστήματος του Κόμβου Πρόσβασης, συμμετέχει στα δύο από τα τρία επίπεδα λειτουργίας του:

1. *το Επίπεδο Χρήστη*: Χειρισμός cells δεδομένων της κίνησης ABR του Κόμβου Πρόσβασης
2. *το Επίπεδο Διαχείρισης*: Εκτέλεση τοπικής διαχείρισης και ανταλλαγή πληροφορίας διαχείρισης με τον κεντρικό επεξεργαστή του Κόμβου Πρόσβασης και εκτέλεση λειτουργιών διαχείρισης.

Η κάρτα Εξυπηρετητή ABR αποτελείται από τα ακόλουθα υποσυστήματα:

1. Το Υποσύστημα Μεταγωγής
2. Τη Μονάδα Εξυπηρέτησης Κίνησης ABR (Μ.Ε.Κ.Α.)
3. Τον Επεξεργαστή Κελιών
4. Τον Εσωτερικό Επεξεργαστή Ελέγχου.

Στην Εικόνα 20 φαίνεται η αντιστοίχιση των υποσυστημάτων της κάρτας με τα δύο επίπεδα λειτουργίας καθώς και με τα επίπεδα του ATM (ATM layers) που επιστρατεύονται για την εκτέλεση αυτών των λειτουργιών.



**Εικόνα 20 Η υλοποίηση των λειτουργιών από τα φυσικά στοιχεία της ABR κάρτας**

Το **Υποσύστημα Μεταγωγής** είναι ουδέτερο όσον αφορά τα επίπεδα λειτουργίας καθώς ο μόνος ρόλος του είναι να διεκπαιρώνει την διεπαφή της κάρτας με τον Κοινό Δίαυλο Κελιών του Κόμβου Πρόσβασης. Το συγκεκριμένο υποσύστημα δέχεται τα cells όλων των τύπων που προέρχονται από τις άλλες κάρτες του συστήματος ή προωθεί cells προς τον Κοινό Δίαυλο Κελιών με προορισμό άλλες κάρτες του συστήματος. Η μόνη λειτουργία του Υποσυστήματος Μεταγωγής είναι η εκτέλεση του πρωτοκόλλου επικοινωνίας των καρτών πάνω από τον δίαυλο και η προώθηση ή αποδοχή των cells προς και από την Μονάδα Εξυπηρέτησης Κίνησης ABR.

Η **Μονάδα Εξυπηρέτησης Κίνησης ABR** εκτελεί λειτουργίες του επιπέδου ATM που αφορούν τον τύπο κίνησης ABR για τα cells δεδομένων. Πιο συγκεκριμένα είναι υπεύθυνη για:

- την αποθήκευση των ABR cells δεδομένων σε ουρές ανά σύνδεση (per flow queueing) μετά την είσοδό τους στον Κόμβο Πρόσβασης
- την χρονοδρομολόγηση (scheduling) των ABR cells δεδομένων κατά την έξοδό τους από τον Κόμβο Πρόσβασης ώστε να επιτυγχάνεται αποδοτική χρήση των πόρων του
- την εκτέλεση λειτουργιών ελέγχου ροής (flow control) μέσω των τεχνικών EFCI marking και Relative Rate Flow Control όπως περιγράφονται στις προδιαγραφές του ATM Forum
- Την προώθηση ή αποδοχή cells Διαχείρισης και RM cells προς και από τον Επεξεργαστή Κελιών τα οποία κατευθύνονται ή προέρχονται από τον Εσωτερικό Επεξεργαστή Ελέγχου.

Ο **Επεξεργαστής Κελιών** αποδέχεται τα cells Διαχείρισης που προέρχονται από τον Κεντρικό Επεξεργαστή και τα RM cells εισόδου από την M.E.K.A. και προωθεί τα cells Διαχείρισης που κατευθύνονται προς τον Κεντρικό Επεξεργαστή και τα επεξεργασμένα RM cells προς την M.E.K.A.

Επίσης έχει την ευθύνη της μεταφοράς όλων των cells του Επιπέδου Διαχείρισης και ειδικών cells ελέγχου ροής (RM cells) του επιπέδου χρήστη προς και από τον Εσωτερικό Επεξεργαστή Ελέγχου. Υλοποιεί τις λειτουργίες αναγνώρισης, προσωρινής αποθήκευσης και μεταφοράς των cells από και προς τον επεξεργαστή, λειτουργίες που εντάσσονται στα χαμηλά στρώματα του επιπέδου ATM.

Τέλος ο Επεξεργαστής Κελιών χρησιμοποιείται επίσης για μερική αποθήκευση και ανάκτηση πληροφορίας που αφορά τις εγκατεστημένες ABR συνδέσεις του Κόμβου Πρόσβασης καθώς και για την εισαγωγή και εξαγωγή της κεφαλίδας δρομολόγησης (CellBus Routing Header) που χρησιμοποιείται για την δρομολόγηση των cells στο κατάλληλο Στοιχείο Μεταγωγής (Cubit) της κάρτας προορισμού μέσω του Κοινού Διαύλου Κελιών (CellBus) (βλ. Κεφάλαιο 1.5.1).

**Εσωτερικός Επεξεργαστής Ελέγχου** είναι υπεύθυνος για:

- την αρχικοποίηση της κάρτας Εξυπηρετητή ABR και τον έλεγχο των υποσυστημάτων της.
- Την υλοποίηση λειτουργιών του επιπέδου ATM που έχουν να κάνουν με την μεταφορά και αποθήκευση cells Διαχείρισης και RM cells που προέρχονται από ή κατευθύνονται προς τον Επεξεργαστή Κελιών
- Την εκτέλεση των λειτουργιών του επιπέδου AAL για τον χειρισμό των πακέτων Διαχείρισης που προέρχονται από τον Κεντρικό Επεξεργαστή του Κόμβου Πρόσβασης. Η λειτουργίες του επιπέδου AAL Segmentation and

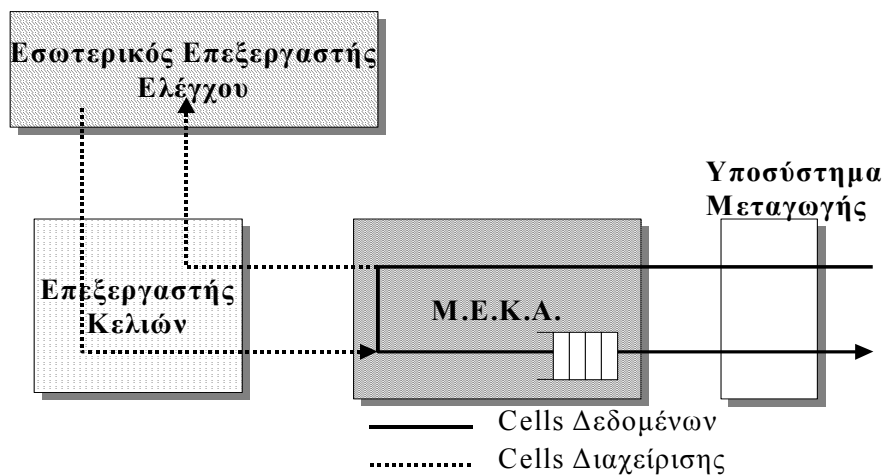


Reassembly (SAR) και Convergence Sublayer (CS) πραγματοποιούνται πάνω στα cells Διαχείρισης που εξάγονται ή εισάγονται στον Επεξεργαστή Κελιών.

- Την υλοποίηση της τοπικής έκδοσης του πρωτοκόλλου επικοινωνίας μεταξύ της κάρτας Εξυπηρετητή Κίνησης ABR και της οντότητας Διαχείρισης του Κεντρικού Επεξεργαστή του Κόμβου Πρόσβασης (βλ. Παράγραφο 1.6.3).
- Την υλοποίηση των εφαρμογών Τοπικής Διαχείρισης (βλ. Παράγραφο 1.5.3)
- Την εκτέλεση των λειτουργιών προγραμματισμού και ενημέρωση της Μ.Ε.Κ.Α.
- Την υλοποίηση του αλγορίθμου ελέγχου ροής του τύπου Explicit Rate Flow Control και την ενημέρωση των εισερχομένων RM cells.

Όλες οι λειτουργίες του Εσωτερικού Επεξεργαστή Ελέγχου είναι υλοποιημένες σε λογισμικό που τρέχει πάνω στον επεξεργαστή και αποτελεί το κύριο θέμα αυτής της εργασίας. Περιγράφεται αναλυτικά στο Κεφάλαιο 3.

Η διαδρομή της κίνησης δεδομένων και διαχείρισης στο εσωτερικό της κάρτας Εξυπηρετητή ABR φαίνεται στην Εικόνα 21



Εικόνα 21 Ροή Κίνησης στο εσωτερικό της κάρτας Εξυπηρετητή ABR

## 2.2 Αρχιτεκτονική της κάρτας Εξυπηρετητή ABR

Η αρχιτεκτονική και η οργάνωση της κάρτας φαίνονται στην Εικόνα 22. Η περιγραφή της δομής, της λειτουργίας και της αλληλεπίδρασης των στοιχείων που αποτελούν την κάρτα περιγράφεται στις παραγράφους αυτού του κεφαλαίου.

## 2.2.1 Το στοιχείο μεταγωγής

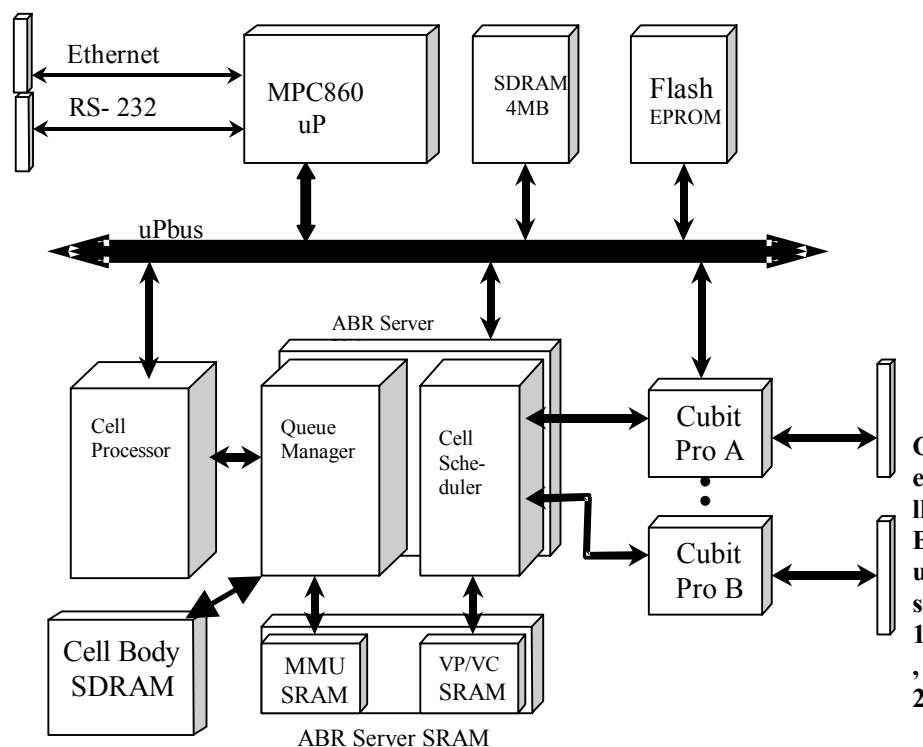
Το στοιχείο μεταγωγής (Transwitch Cubit Pro)[9] συνδέεται με την Μονάδα Εξυπηρέτησης ABR (ABR Server Unit), μέσω μιας σύνδεσης UTOPIA και με το Backplane μέσω των 37 γραμμών του CellBus. Ακόμη συνδέεται με τον επεξεργαστή MPC860, μέσω μιας διασύνδεσης επεξεργαστή (με επιλεγμένη την κατάσταση Motorola) πάνω από το μP Bus με 16 γραμμές διεύθυνσης (address lines) και 8 γραμμές δεδομένων (data lines). Κάθε κάρτα εξυπηρέτησης ABR διαθέτει 2 Cubit Pro, με την κάθε μια να είναι συνδεδεμένη σε ένα από τα δύο CellBus. Μόνο ένα CubitPro είναι σε λειτουργία μια δεδομένη στιγμή ενώ το άλλο είναι σε αναμονή για την περίπτωση αστοχίας του πρωτεύοντος CellBus. Σε αυτή τη περίπτωση το εν αναμονή CubitPro τίθεται σε λειτουργία ενώ αυτό του πρωτεύοντος CellBus απενεργοποιείται. Το Cubit-Pro δέχεται Cells από το CellBus και τα προωθεί στη Μονάδα Εξυπηρέτησης

ABR μέσω μιας ουράς FIFO 123 cells. Στην αντίθετη κατεύθυνση αποδέχεται cells από τη στη Μονάδα Εξυπηρέτησης ABR και τα προωθεί στο CellBus μέσω μιας ουράς FIFO 4 cells. Η συσκευή CubitPro παρέχει μαρκάρισμα ένδειξης Συνοστισμού (Congestion indicator marking) και πέταγμα cells (cell discarding) κάτω από ορισμένες συνθήκες.

## 2.2.2 Η Μονάδα Εξυπηρέτης Κίνησης ABR (M.E.K.A.)

Η μονάδα εξυπηρέτησης κίνησης ABR (ABR Server Unit – ABRSU) αποτελείται από 2 στοιχεία, τον προγραμματιστή Cell (Cell Scheduler) και τον διαχειριστή ουρών (Queue Manager). Συνδέεται με τις συσκευές CubitPro μέσω μιας διασύνδεσης UTOPIA και με τον επεξεργαστή MPC860. Μια μονάδα SDRAM είναι συνδεδεμένη με την ABRSU με σκοπό την αποθήκευση Cells και 2 ( ή 1) μονάδες SRAM για αποθήκευση πινάκων αποθήκευσης και δόμησης ουρών. Η λειτουργία της ABRSU συνίσταται στη αποθήκευση των cells τύπου ABR με βάση το VP/VC τους (εισερχόμενη κατεύθυνση) και στον προγραμματισμό μετάδοσης των cells (εξερχόμενη διεύθυνση). Το σύνολο της λογικής της μονάδας εξυπηρέτησης κίνησης ABR υλοποιείται από μια FPGA συνδεδεμένη με μνήμες για την αποθήκευση των δεδομένων και των εσωτερικών δομών (head-tail pointers για τις ουρές, κ.α.)

Η μονάδα εξυπηρέτησης κίνησης ABR πρέπει να υποστηρίζει αρκετές ουρές ώστε να αποφευχθούν προβλήματα «head-of-line blocking», και να υποστηρίζει αρκετό αποθηκευτικό χώρο για τα δεδομένα των cells ώστε να μπορεί να απορροφά διακυμάνσεις στην διαθεσιμότητα bandwidth.



Εικόνα 22 Οργάνωση της κάρτας Εξυπηρετητή ABR

Αναλυτική περιγραφή της σχεδίασης της Μ.Ε.Κ.Α. βρίσκεται στο Παράρτημα Α

### 2.2.3 Ο Επεξεργαστής Κελιών

Ο Επεξεργαστής Κελιών (Motorola MC92501 Cell Processor) [7] συνδέεται με την Μ.Ε.Κ.Α. μέσω μίας διεπαφής Standard UTOPIA 2 μέσω της οποίας αποδέχεται ή αποστέλλει cells Διαχείρισης και RM cells προς την Μ.Ε.Κ.Α. Επίσης συνδέεται με τον Εσωτερικό Επεξεργαστή Ελέγχου μέσω εσωτερικής διεπαφής μικροεπεξεργαστή με χρήση καταχωρητών (Registered Memory mapped interface) πάνω στο μP Bus με 24 γραμμές διεύθυνσης (address lines) και 32 γραμμές δεδομένων (data lines). Ο Επεξεργαστής Κελιών χρησιμοποιεί μία SRAM η οποία είναι συνδεδεμένη στην εσωτερική διεπαφή μνήμης του, προκειμένου να αποθηκεύει πληροφορία που αφορά τις συνδέσεις που εξυπηρετεί και την εσωτερική λειτουργία του.

Ο Επεξεργαστής Κελιών είναι προγραμματισμένος έτσι ώστε να δέχεται cells από την Μ.Ε.Κ.Α., να αφαιρεί τις επιπλέον κεφαλίδες των cells (Tandem-Routing Header) και να μεταφέρει όλα τα cells που δέχεται στον Εσωτερικό Επεξεργαστή Ελέγχου. Η μεταφορά των cells γίνεται μέσω της διεπαφής του Επεξεργαστή Κελιών με τον Εσωτερικό Επεξεργαστή. Τα cells που πρόκειται να εξαχθούν προς τον Εσωτερικό Επεξεργαστή αποθηκεύονται σε μία εσωτερική FIFO ουρά μήκους 16 cells η οποία διατηρείται στο εσωτερικό του Επεξεργαστή Κελιών. Όταν η ουρά εξόδου του Επεξεργαστή Κελιών παύει να είναι κενή (δηλαδή όταν έχουν φτάσει cells που κατευθύνονται προς τον Εσωτερικό Επεξεργαστή) τότε παράγεται μία διακοπή (interrupt) η οποία ενημερώνει τον Εσωτερικό Επεξεργαστή ότι υπάρχουν cells τα

οποία πρέπει να διαβάσει. Αυτή η ενημέρωση γίνεται μέσω ενός ειδικού Καταχωρητή Διακοπών (Interrupt Register) ο οποίος βρίσκεται εωτερικά στον Επεξεργαστή Κελιών.

Στην αντίθετη κατεύθυνση, ο Επεξεργαστής Κελιών είναι προγραμματισμένος να δέχεται cells Διαχείρισης και RM cells από τον Εσωτερικό Επεξεργαστή Ελέγχου, να τοποθετεί τις απαραίτητες επιπλέον κεφαλίδες των cells (Tandem-Routing Header) και να προωθεί τα cells προς την Μ.Ε.Κ.Α. μέσω της διεπαφής UTOPIA. Η πληροφορία των επιπλέον κεφαλίδων για κάθε σύνδεση καθώς και για την εσωτερική σύνδεση με την Κάρτα Κεντρικού Επεξεργαστή, διατηρείται στην SRAM του Επεξεργαστή Κελιών και ενημερώνεται κάθε φορά που στην Κάρτα Εξυπηρετητή Κίνησης ABR εγκαθίσταται ή ελευθερώνεται μία σύνδεση ABR.

Ο Επεξεργαστής Κελιών δέχεται cells μέσω της διεπαφής του με τον Εσωτερικό Επεξεργαστή. Τα cells που πρόκειται να εισαχθούν από τον Εσωτερικό Επεξεργαστή τοποθετούνται ένα κάθε φορά σε έναν εσωτερικό cell buffer μήκους ενός cell. Όταν ο cell buffer είναι κενός και μπορεί να δεχθεί ένα νέο cell από τον Εσωτερικό Επεξεργαστή τότε ο Επεξεργαστής Κελιών παράγει μία διακοπή (interrupt) η οποία ενημερώνει τον Εσωτερικό Επεξεργαστή ότι μπορεί να εισάγει ένα νέο cell. Αυτή η ενημέρωση γίνεται μέσω του ειδικού Καταχωρητή Διακοπών (Interrupt Register).

Τέλος να σημειωθεί ότι ο Επεξεργαστής Κελιών δεν πραγματοποιεί καμία μετάφραση διευθύνσεων στο πεδίο VP/VC αλλά διατηρεί στο πεδίο αυτό το Εσωτερικό Ενδεικτικό Σύνδεσης (Internal Connection Identifier) το οποίο έχει εισαχθεί από την κάρτα (ATM ή VDSL) εισόδου κατά την διάρκεια της μετάφρασης που έχει γίνει εκεί.

## 2.2.4 Εσωτερικός Επεξεργαστής Ελέγχου

Ο επεξεργαστής που χρησιμοποιείται για τον εσωτερικό έλεγχο της κάρτας είναι MPC860 [8] και βασίζεται στην αρχιτεκτονική PowerQUICC της Motorola. Ο επεξεργαστής εκτελεί το λογισμικό υψηλού επιπέδου που αφορά την Διαχείριση της κάρτας, την υλοποίηση πρωτοκόλλων για την επικοινωνία με τια άλλες κάρτες του συστήματος και την εκτέλεση αλγορίθμων για τον έλεγχο ροής της κίνησης ABR. Επίσης εκτελεί λογισμικό χαμηλού επιπέδου για την αλληλεπίδραση με τα υπολοιπα στοιχεία της κάρτας Εξυπηρετητή ABR. Αναλυτική περιγραφή των λειτουργιών του μικροεπεξεργαστή και του λογισμικού του γίνεται στο Κεφάλαιο 3. Στην συνέχεια περιγράφεται η τοποθέτηση του επεξεργαστή μέσα στο σύστημα και η αλληλεπίδραση του με τα στοιχεία της κάρτας καθώς και με τα περιφερειακά του σε επίπεδο αρχιτεκτονικής

Ο MPC860 συνδέεται με τις διάφορες μονάδες της κάρτας καθώς και με τα περιφερειακά του μέσω ενός κοινού διαύλου uPBus 64 γραμμών (32 γραμμές διεύθυνσης-32 γραμμές δεδομένων).

Χρησιμοποιεί την διασύνδεση μικροεπεξεργαστή της συσκευής Επεξεργαστή Κελιών (MC92501) για να τον αρχικοποιήσει, να τον προγραμματίσει και για να ανταλλάξει cells Διαχείρισης (Management Cells) με τον μικροεπεξεργαστή που βρίσκεται στη

κάρτα Κεντρικού Επεξεργαστή καθώς και για να δεχθεί και να αποστείλει RM cells των συνδέσεων ABR που εξυπηρετούνται από την κάρτα.

Ακόμη, ο MPC860 διασυνδέεται με την Μ.Ε.Κ.Α. μέσω της διεπαφής της προκειμένου να την αρχικοποιήσει εσωτερικά και να αρχικοποιήσει την μνήμη αποθήκευσης των ABR cells. Στην κανονική λειτουργία ο μικροεπεξεργαστής ενημερώνει την Μ.Ε.Κ.Α. για την εγκατάσταση νέων συνδέσεων ή για την παύση παλιών, και ταυτόχρονα της παρέχει απαραίτητες πληροφορίες για τον χειρισμό των ABR cells όπως η κατάσταση των πόρων του συστήματος (διαθέσιμο εύρος ζώνης στην έξοδο) και το μήκος των ουρών αποθήκευσης των ABR cells πάνω από το οποίο η Μ.Ε.Κ.Α θα πρέπει να θεωρεί ότι υφίσταται συμφόριση στο σύστημα. Επίσης ο επεξεργαστής μέσω της ίδιας διεπαφής ενημερώνεται για την κατάσταση των συνδέσεων εσωτερικά της Μ.Ε.Κ.Α. και μπορεί να έχει πρόσβαση στην εξωτερική μνήμη της Μ.Ε.Κ.Α. προκειμένου να προσπελάσει χρήσιμες πληροφορίες για την διαχείρισή της.

Επιπλέον ο MPC860 είναι υπεύθυνος για την αρχικοποίηση και τον προγραμματισμό του Στοιχείου Μεταγωγής (CubitPro). Το CubitPro με την σειρά του παρέχει πληροφορίες για την κίνηση και στατιστικά, χρήσιμα για την διαχείριση του όλου συστήματος. Ακόμη σε περίπτωση ελαττωματικής λειτουργίας του ενεργού CellBus, ο MPC860 είναι υπεύθυνος για την απενεργοποίησή του και την ενεργοποίηση του αναπληρωματικού.

Ο επεξεργαστής MPC860 χρησιμοποιεί μια μονάδα Flash EPROM που αποθηκεύει το λειτουργικό του σύστημα καθώς και δεδομένα για την εκκίνηση και την αρχικοποίηση του. Επιπλέον χρησιμοποιεί μια μνήμη SDRAM για αποθήκευση δεδομένων του λογισμικού και cells. Λόγω του εσωτερικού ελεγκτή μνήμης (on-chip memory controller), ο MPC860 μπορεί να συνδεθεί με τα παραπάνω χωρίς επιπλέον λογική κετευθείαν πάνω στο μP bus.

Τέλος, για τον έλεγχο του συστήματος, ο επεξεργαστής συντηρεί μια σύνδεση RS-232/432 που ελέγχεται και αυτή μέσω ενός σειριακού ελεγκτή πάνω στο ολοκληρωμένο του MPC860 καθώς και μία σύνδεση Ethernet για εξωτερική δικτυακή σύνδεση του (και κατ'επέκταση δικτυακή σύνδεση της κάρτας) με τον έξω κόσμο. Η λογική διεπαφής Ethernet βρίσκεται και αυτή εσωτερικά (on-chip) στον επεξεργαστή οπότε δεν χρειάζονται επιπλέον εξωτερικά στοιχεία για την δικτυακή σύνδεση.

### **3 Το λογισμικό της εξυπηρετητή ABR**

#### **3.1 Περιγραφή Λειτουργίας**

Ο Εσωτερικός Επεξεργαστής Ελέγχου (MPC860) της Κάρτας Εξυπηρετητή ABR είναι ένας επεξεργαστής γενικού σκοπού που ανήκει στην αρχιτεκτονική (Power Pc της Motorola). Ως γενικού σκοπού επεξεργαστής χρησιμοποιείται στην κάρτα ως «ενσωματωμένος» (embedded) και η γενική χρήση του είναι η εκτέλεση σε λογισμικό όλων των λειτουργιών που απαιτούνται από το σύστημα και η πολυπλοκότητα των

οποίων καθιστά ανέφικτη την υλοποίηση τους σε υλικό. Επίσης ο επεξεργαστής παίζει τον ρόλο του εγκεφάλου του συστήματος καθώς αρμοδιότητα του είναι η αρχικοποίηση, εκκίνηση και έλεγχος των λειτουργιών του υλικού μέρους της κάρτας, λειτουργίες που προφανώς δεν μπορούν εκτελεστούν από τα ίδια τα υλικά στοιχεία προς τους εαυτούς τους (αυτοπρογραμματισμός).

Στο υψηλότερο επίπεδο αφαίρεσης, οι εργασίες που εκτελεί ο «ενσωματωμένος» επεξεργαστής της κάρτας Εξυπηρετητή ABR είναι δύο, η *Διαχείριση (Management)* και ο *Έλεγχος Ροής (Flow Control)*.

### 3.1.1 Διαχείριση

Η λειτουργία της Διαχείρισης αφορά την Τοπική Διαχείριση της κάρτας όπως αυτή περιγράφηκε στο κεφάλαιο 1. Τα μέρη αυτής της λειτουργίας περιγράφονται στη συνέχεια.

- Αρχικοποίηση και προγραμματισμός της ABR κάρτας. Πραγματοποιείται κατά την αρχικοποίηση του συστήματος ή μετά από απαίτηση της κεντρικής CPU. Η αρχικοποίηση ξεκινά με την εκκίνηση της CPU (booting) και downloading των προγραμμάτων. Τα chips της κάρτας ABR (FPGA, Cell Processor και CUBIT) αρχικοποιούνται, προγραμματίζονται και ελέγχονται. Στο τέλος η τοπική CPU αναφέρει την παρουσία της και την ταυτότητά της στην κεντρική CPU.
- Διαχείριση Συνδέσεων. Αυτή η λειτουργία αφορά την ανημέρωση των φυσικών στοιχείων της κάρτας και κυρίως της M.E.K.A. με τις απαραίτητες παραμέτρους (Flow Id, Flow Group Id, Available Output Bandwidth, Minimum Connection Cell Rate κ.τ.λ.) κάθε φορά που εγκαθίσταται μία νέα ή κλείνει μία παλιά σύνδεση. Επίσης αφορά την ενημέρωση εσωτερικών παραμέτρων που απαιτούνται για την εκτέλεση λειτουργιών επί των συνδέσεων. Η απαραίτητη αυτή πληροφορία προέρχεται από το Κεντρικό Επίπεδο Διαχείρισης που βρίσκεται στην κάρτα Κεντρικού Επεξεργαστή, και μεταφέρεται στην κάρτα Εξυπηρετητή ABR μέσω της επικοινωνίας μεταξύ των δύο καρτών.
- Έλεγχος chips. Η τοπική CPU προγραμματίζει τα chips της κάρτας, ελέγχει περιοδικά την κατάσταση τους και συλλέγει στατιστικά στοιχεία που αφορούν τις ABR συνδέσεις. Αυτός ο έλεγχος πραγματοποιείται μέσω λειτουργιών που παρέχει η διεπαφή του κάθε στοιχείου της κάρτας με τον Επεξεργαστή Ελέγχου.
- Διαχείριση Σφαλμάτων. Ελέγχονται πιθανά σφάλματα και είτε διορθώνονται είτε απομονώνονται και αναφέρονται στην κεντρική CPU. Αυτά τα σφάλματα αναφέρονται στον Επεξεργαστή Ελέγχου μέσω των διεπαφών των στοιχείων της κάρτας με αυτόν.
- Επικοινωνία με την κεντρική CPU. Η τοπική CPU εξηηρετεί τις αιτήσεις της κεντρικής CPU για ανταλλαγή δεδομένων και εκτέλεση εντολών.. Το πρωτόκολλο ICP είναι υπεύθυνο για την επικοινωνία υψηλού επιπέδου όπως μεταφορά πληροφορίας για την κατάσταση της ABR κάρτας και εντολών CAC. Αυτή η επικοινωνία απαιτεί από το λογισμικό στους επεξεργαστές των καρτών να

υλοποιούν και στρώματα (layers) ATM τα οποία συνήθως υλοποιούν τελικοί χρήστες δικτύων ATM όπως το στρώμα AAL και το στρώμα ATM, καθώς και πρωτόκολλα υψηλότερου επιπέδου για την επεξεργασία των πακέτων πληροφορίας που μεταφέρονται μέσω του εσωτερικού δικτύου ATM.

### 3.1.2 Έλεγχος Ροής

Η κάρτα ABR χειριζόμενη την ABR κυκλοφορία είναι αναγκαίο να υποστηρίζει μία από τις βασικότερες λειτουργίες της υπηρεσίας ABR που είναι ο Έλεγχος Ροής. (βλ. Παράρτημα A.1).

Ο Έλεγχος Ροής (Flow Control) όπως ορίζεται από τα πρότυπα του ATM Forum έχει δύο μορφές [4]:

- Relative Rate Flow Control
- Explicit Rate Flow Control

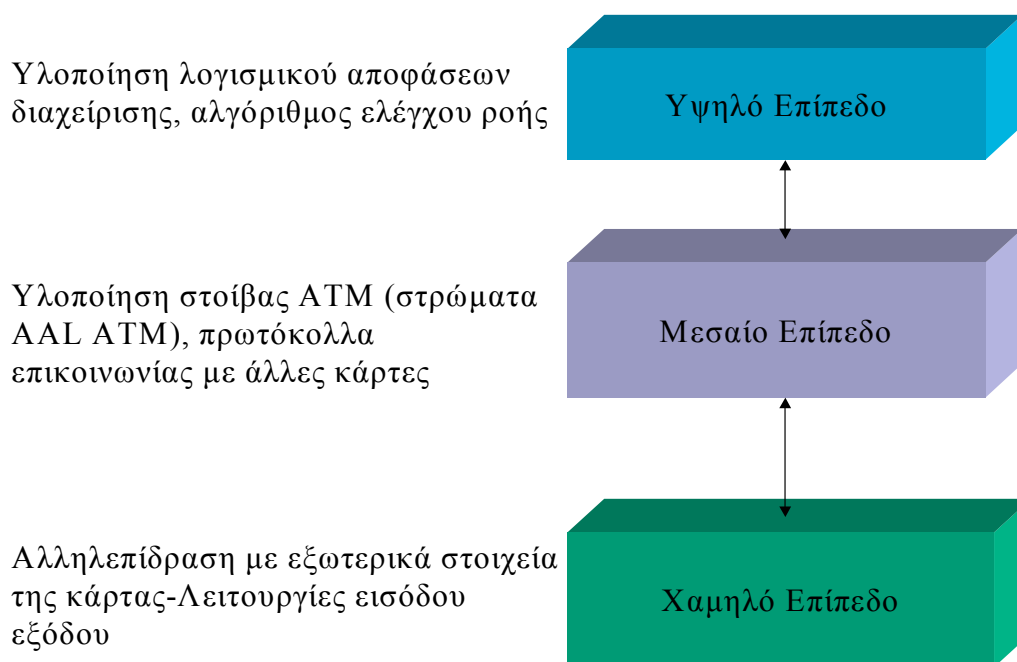
Η πρώτη μορφή λόγω της απλότητας της υλοποίησης της είναι δυνατόν να υλοποιηθεί σε υλικό. Η δεύτερη μορφή απαιτεί υψηλού επιπέδου και μεγάλης πολυπλοκότητας αλγόριθμους οι οποίοι είναι απαραίτητο να υλοποιούνται από το λογισμικό της κάρτας.

Αυτού του είδους ο έλεγχος ροής έχει ενσωματωθεί στη σχεδίαση του λογισμικού της ABR κάρτας με όσο το δυνατόν πιο ευέλικτη μορφή. Η σχεδίαση της συγκεκριμένης λειτουργικότητας βασίζεται σε ένα γενικό υπόβαθρο χειρισμού των ειδικών cells που μεταφέρουν την πληροφορία ελέγχου ροής (RM cells) από τον επεξεργαστή μέσω των διεπαφών των φυσικών στοιχείων (Επεξεργαστής Κελιών).

Πάνω από αυτό το υπόβαθρο είναι δυνατόν να υλοποιηθεί ένας μεγάλος αριθμός αλγόριθμων Explicit Rate Flow Control, που βασίζονται αποκλειστικά στην πληροφορία που μεταφέρεται από τα RM cells, με ανεξάρτητο τρόπο.

## 3.2 Αρχιτεκτονική Λογισμικού της ABR κάρτας

Το λογισμικό της κάρτας Εξηγηρητητή ABR είναι οργανωμένο σε τρία ιεραρχικά επίπεδα. Όπως φαίνεται στην Εικόνα 23



Εικόνα 23 Ιεραρχικά Επίπεδα Λογισμικού κάρτας Εξυπηρετητή ABR

### 3.2.1 Χαμηλό Επίπεδο

Το χαμηλό επίπεδο περιέχει όλες τις λειτουργίες αλληλεπίδρασης του επεξεργαστή με τα στοιχεία της κάρτας. Αυτές οι λειτουργίες είναι απαραίτητες για τα υψηλότερα επίπεδα καθώς παρέχουν σε αυτά τις κατάλληλες διεπαφές ανταλλαγής πληροφορίας με το εξωτερικό σύστημα. Πιο συγκεκριμένα οι λειτουργίες αυτού του επιπέδου είναι υπεύθυνες για

- ✓ την μεταφορά ενός cell από τον Επεξεργαστή Κελιών στο εσωτερικό του Επεξεργαστή
- ✓ την εγγραφή στις διεπαφές όλων των στοιχείων της κάρτας των τιμών των παραμέτρων της αρχικοποίησης τους
- ✓ την εγγραφή των εντολών της Μ.Ε.Κ.Α. στην και την ανάγνωση δεδομένων από την διεπαφή της με τον επεξεργαστή
- ✓ τέλος για την πρόσβαση στην εξωτερική μνήμη των στοιχείων της κάρτας καθώς και για την προσπέλαση χρήσιμης πληροφορίας που αφορά την κατάσταση του συστήματος και είναι αποθηκευμένη με διάφορους τρόπους στα στοιχεία που δομούν το σύστημα.

Ο επεξεργαστής ως ενσωματωμένο στοιχείο ελέγχου πρέπει να αλληλεπιδρά σε χαμηλό επίπεδο απευθείας με το υλικό των υπόλοιπων στοιχείων της κάρτας καθώς και με τις μνήμες του. Πρέπει συνεπώς να εκτελεί απλό λογισμικό διεπαφών το οποίο θα γνωρίζει όλες τις λεπτομέρειες τις λειτουργίας και της δομής του υλικού της κάρτας και κυρίως τις λεπτομέρειες των διεπαφών των στοιχείων με τον



επεξεργαστή. Αυτό το λογισμικό αφενός θα κωδικοποιεί σε μορφή κατανοητή από τον επεξεργαστή τα μέρη εκείνα του λογισμικού με τα οποία αλληλεπιδρά και αφετέρου θα τρέχει όλες τις απαραίτητες διαδικασίες ώστε ο επεξεργαστής να παρέχει ή να δέχεται χρήσιμη πληροφορία απευθείας από το υλικό των στοιχείων.

### **3.2.1.1 Αλληλεπίδραση επεξεργαστή με το υλικό (Hardware) της κάρτας.**

Όπως γίνεται στην πλειονότητα των embedded συστημάτων έτσι και στην Κάρτα Εξυπηρετητή ABR ο embedded Επεξεργαστής Ελέγχου αλληλεπιδρά με τα το υλικό μέρος του συστήματος (τα διάφορα στοιχεία που υλοποιούν την κάρτα) μέσω των διεπαφών επεξεργαστή (micro processor interface) που αυτά έχουν. Αυτές οι διεπαφές καθορίζονται από τους κατασκευαστές των στοιχείων και διαφέρουν ανάλογα με την λειτουργικότητα του στοιχείου. Ωστόσο η γενική λογική υλοποίησης των διεπαφών αυτών είναι η ίδια για όλα τα στοιχεία που προορίζονται για embedded λειτουργία..

Γενικά σε ένα embedded σύστημα όπως αυτό της κάρτας Εξυπηρετητή ABR, τα στοιχεία που το υλοποιούν συνδέονται με τον επεξεργαστή μέσω ενός κοινού διαύλου (μP Bus). Σε κάθε ένα από αυτά τα στοιχεία είναι υλοποιημένη εσωτερικά μία διεπαφή με τον επεξεργαστή η οποία αποτελείται από μία ομάδα καταχωρητών (μP interface registers) οι οποίοι είναι εγγράψιμοι ή αναγνώσιμοι από τον επεξεργαστή μέσω του κοινού διαύλου. Αυτοί οι καταχωρητές περιέχουν πληροφορία την οποία χρησιμοποιούν τα στοιχεία για να καθορίσουν την εσωτερική λειτουργία τους (εσωτερική πληροφορία), ή πληροφορία η οποία προορίζεται για επεξεργασία από τον επεξεργαστή (εξωτερική πληροφορία) δηλαδή για τις διαδικασίες εισόδου/εξόδου (I/O) του επεξεργαστή. Ο επεξεργαστής προσπελαύνει αυτούς του καταχωρητές για να ενημερωθεί (ανάγνωση) ή να καθορίσει (εγγραφή) την εσωτερική λειτουργία του στοιχείου στην περίπτωση της εσωτερικής πληροφορίας, ενώ προσπελαύνει τους καταχωρητές προκειμένου να δεχθεί πληροφορία που είναι προς επεξεργασία από τον ίδιο (εγγραφή) ή να εξάγει πληροφορία που έχει ήδη επεξεργαστεί στην περίπτωση εξωτερικής πληροφορίας.

Η προσπέλαση των εσωτερικών καταχωρητών διεπαφής των διαφόρων στοιχείων της κάρτας από τον επεξεργαστή γίνεται όμοια με την προσπέλαση της μνήμης. Πιο συγκεκριμένα ο επεξεργαστής τοποθετεί την διεύθυνση του καταχωρητή που θέλει να προσπελάσει καθώς και τα δεδομένα, πάνω στον κοινό δίαυλο του οποίου είναι και ο ελεγκτής. Επειδή η εσωτερικές διευθύνσεις των καταχωρητών των διαφόρων στοιχείων είναι κοινές, ο επεξεργαστής προκειμένου να καθορίσει το στοιχείο του οποίου τους καταχωρητές θέλει να προσπελάσει χρησιμοποιεί κάποια συγκεκριμένα σήματα τα οποία είναι έξοδοι του επεξεργαστή και είσοδοι των άλλων στοιχείων. Αυτά τα σήματα ονομάζονται Chip Select (CS). Το κάθε στοιχείο της κάρτας συνδέεται μέσω ενός ξεχωριστού CS με τον επεξεργαστή. Ανάλογα με το στοιχείο το οποίο θέλει να προσπελάσει ο επεξεργαστής, ταυτόχρονα με την διεύθυνση του εσωτερικού καταχωρητή του στοιχείου, οδηγεί και το αντίστοιχο CS σήμα..

Προκειμένου ο επεξεργαστής να αλληλεπιδρά σωστά με τα διάφορα embedded στοιχεία θα πρέπει να διατηρεί εσωτερικά μία εικόνα των διεπαφών τους. Αυτή η εικόνα αποτελείται από τις διευθύνσεις των διαφόρων καταχωρητών διεπαφής των

στοιχείων. Οι διευθύνσεις που αντιστοιχεί εσωτερικά ο επεξεργαστής για τους διάφορους καταχωρητές των εξωτερικών στοιχείων δεν μπορούν να συμπίπτουν με τις απόλυτες εσωτερικές διευθύνσεις που έχουν αναθέσει οι κατασκευαστές των στοιχείων στους καταχωρητές τους καθώς αυτές μπορεί να είναι κοινές για δύο διαφορετικά στοιχεία και συνεπώς υπάρχει ο κίνδυνος της αναντιστοιχίας.

Για αυτόν τον λόγο ο επεξεργαστής διατηρεί εσωτερικά ένα μεγάλο σύνολο διευθύνσεων που ονομάζεται χώρος διευθύνσεων (memory space) στις οποίες αντιστοιχεί εν έκάστο από τους καταχωρητές των διεπαφών των εξωτερικών στοιχείων. Αυτή η λειτουργία ονομάζεται αντιστίχιση διευθύνσεων (memory map). Οι εσωτερικές διευθύνσεις έχουν μήκος μεγαλύτερο από το μήκος των διευθύνσεων του κοινού διαύλου καθώς το μέγεθος του χώρου διευθύνσεων πρέπει να είναι αρκετά μεγάλο ώστε να καλύπτει όλους τους καταχωρητές διεπαφής όλων των στοιχείων του συστήματος. Το κομμάτι του χώρου διευθύνσεων που αντιστοιχείται σε κάθε στοιχείο προκύπτει από το σύνολο των εσωτερικών διευθύνσεων των καταχωρητών διεπαφής εισάγοντας επιπλέον bits στην αρχή τους. Ο αριθμός αυτών των bits είναι ίσος με την διαφορά του αριθμού των bits των διευθύνσεων που χρησιμοποιεί ο επεξεργαστής για τον δικό του χώρο διευθύνσεων από τον αριθμό των bits που χρησιμοποιούν τα στοιχεία εσωτερικά προκειμένου να διευθυνσιοδοτήσουν τους καταχωρητές τους. Με αυτά τα επιπλέον bits ο επεξεργαστής κατορθώνει να αντιστοιχίσει μοναδικά τον κάθε καταχωρητή του κάθε στοιχείου με μία ξεχωριστή διεύθυνση.

Κάθε φορά που το λογισμικό θέλει να προσπελάσει έναν καταχωρητή διεπαφής ενός στοιχείου αναφέρεται σε αυτό με την αντιστοιχισμένη (memory mapped) διεύθυνση. Στη συνέχεια το υλικό του επεξεργαστή είναι υπεύθυνο, με βάση την αντιστοιχισμένη διεύθυνση, να τοποθετήσει στον κοινό διάυλο την απόλυτη διεύθυνση του καταχωρητή (αφού αφαιρέσει τα επιπλέον bits) και να οδηγήσει το κατάλληλο CS σήμα το οποίο θα ενεργοποιήσει την διεπαφή του στοιχείου στο οποίο ανήκει ο προς προσπέλαση καταχωρητής, απαγορεύοντας στα υπόλοιπα στοιχεία να χρησιμοποιήσουν την διεύθυνση του κοινού διαύλου. Προκειμένου το υλικό του επεξεργαστή να εκτελέσει την παραπάνω λειτουργία πρέπει βεβαίως να διατηρεί εσωτερικά και μία αντιστοίχιση των κομματιών του χώρου διευθύνσεων που διατίθενται για κάθε ένα στοιχείο με τα αντίστοιχα CS σήματα του. Η αντιστοίχιση διευθύνσεων πραγματοποιείται κατά την αρχικοποίηση του επεξεργαστή. Τέλος το υλικό του επεξεργαστή είναι υπεύθυνο να υλοποιήσει τα πρωτόκολλα επικοινωνίας με τα στοιχεία της κάρτας πάνω από τον διάυλο (Bus) και τους κατάλληλους χρονισμούς. Οι παράμετροι που καθορίζουν τις διαδικασίες και τους χρονισμούς για την επικοινωνία των διεπαφών ορίζονται επίσης κατά την αρχικοποίηση του επεξεργαστή.

### **3.2.1.2 Αντιστοίχιση Διευθύνσεων (Memory Map)**

Ο επεξεργαστής της κάρτας Εξυπηρετητή Κίνησης ABR MPC860 έχει 8 προγραμματιζόμενα CS σήματα, τα οποία μπορούν να χρησιμοποιηθούν για την πρόσβαση στα εξωτερικά στοιχεία της κάρτας. Στην συγκεκριμένη εφαρμογή ο χώρος διευθύνσεων του μικροεπεξεργαστή χωρίζεται σε 8 μέρη και καθορίστηκε όπως φαίνεται στον Πίνακας 4.

CSx	Περιοχή μνήμης	Στοιχείο
Cs0	0x00000000 0x001FFFFFFF	FLASH memory
-	0x002000000 0x1FFFFFFF	Αχρησιμοποίητο
Cs1	0x20000000 0x203E8000	SDRAM memory (Block 1)
-	0x203E8001 0x3FFFFFFF	Αχρησιμοποίητο
Cs2	0x40000000 0x403E8000	SDRAM memory (Block 2)
-	0x403E8001 0x5FFFFFFF	Αχρησιμοποίητο
Cs3	0x60000000 0x600000FF	CUBIT 1
-	0x60000100 0x7FFFFFFF	Αχρησιμοποίητο
Cs4	0x80000000 0x900000FF	CUBIT 2
-	0x90000100 0x9FFFFFFF	Αχρησιμοποίητο
Cs5	0xA0000000 0xA0000080	Αχρησιμοποίητο
-	0xA0000081 0xBFFFFFFF	Αχρησιμοποίητο
Cs6	0xC0000000 0xC3FFFFFFF	MC92501
-	0xC4000000 0xCFFFFFFF	Unused
CS7	0xD0000000 0xD000000F	M.E.K.A.
-	0xD0000010 0xFFFFFFFF	Unused

Πίνακας 4

### 3.2.1.3 Συγχρονισμός μικροεπεξεργαστή με τα στοιχεία της κάρτας

Ένα μεγάλο μέρος των λειτουργιών που επιτελεί ο Εσωτερικός Επεξεργαστής Ελέγχου, είναι λειτουργίες Πραγματικού Χρόνου (Real Time Operations). Πιο συγκεκριμένα οι σημαντικές εργασίες του επεξεργαστή είναι ευαίσθητες σε χρονικούς περιορισμούς και συνεπώς πρέπει να εκτελούνται με τέτοιο τρόπο ώστε αυτοί οι περιορισμοί να μην καταπατώνται γεγονός που μπορεί να οδηγήσει στην απορύθμιση του συστήματος, στην επεξεργασία λανθασμένης πληροφορίας και συνεπώς στην τελική πτώση του. Σε αυτήν την λειτουργικότητα πραγματικού χρόνου μεγάλο μέρος ευθύνης φέρουν οι διαδικασίες χαμηλού επιπέδου που αφορούν την αλληλεπίδραση του μικροεπεξεργαστή με τα στοιχεία της κάρτας, και κυρίως το

κομμάτι εκείνο της αλληλεπίδρασης που έχει να κάνει με τον συγχρονισμό της λειτουργίας των στοιχείων της κάρτας με τις λειτουργίες του επεξεργαστή.

Στα συστήματα πραγματικού χρόνου, υπάρχουν δύο τεχνικές συγχρονισμού του επεξεργαστή με τα εξωτερικά στοιχεία, το polling και οι διακοπές (interrupts).

Με το polling ο επεξεργαστής εκτελεί βρόγχους (loops) ελέγχου των διεπαφών των στοιχείων. Αυτοί οι βρόγχοι εκτελούνται είτε συνεχώς είτε ανα συγκεκριμένα χρονικά διαστήματα, και κάθε φορά ελέγχουν αν τα στοιχεία με τα οποία αλληλεπιδρά ο επεξεργαστής είναι έτοιμα να επικοινωνήσουν μαζί του. Έτσι αν επεξεργαστής θέλει να γράψει πληροφορία ελέγχει τακτικά το πότε τα στοιχεία είναι έτοιμα να την δεχθούν, ενώ ελέγχει επίσης τακτικά το αν τα στοιχεία θέλουν να μεταφέρουν πληροφορία προς τον επεξεργαστή. Και στις δύο περιπτώσεις εκτελεί τις λειτουργίες ανάγνωσης ή εγγραφής αμέσως μόλις αντιληφθεί ότι τα στοιχεία είναι έτοιμα.

Στην των διακοπών, ο επεξεργαστής δεν έχει την ευθύνη να ενημερώνεται μόνος του για τις λειτουργίες αλληλεπίδρασης. Αντίθετα όταν τα στοιχεία της κάρτας είναι έτοιμα να δεχθούν πληροφορία ή έχουν πληροφορία η οποία πρέπει να μεταφερθεί στον επεξεργαστή, τότε παράγουν σήματα διακοπών τα οποία ειδοποιούν τον επεξεργαστή ο οποίος με την σειρά του αναλαμβάνει να εξυπηρετήσει τα στοιχεία. Το κάθε στοιχείο της κάρτας συνδέεται με τον επεξεργαστή με ένα διαφορετικά σήμα διακοπής έτσι ώστε ο δεύτερος να γνωρίζει από ποιο στοιχείο προέρχεται η ειδοποίηση. Επίσης το κάθε στοιχείο περιέχει στην διεπαφή του με τον επεξεργαστή έναν καταχωρητή που ονομάζεται καταχωρητής διακοπών (Interrupt Register) ο οποίος περιέχει την απαραίτητη πληροφορία που αφορά τους λόγους για τους οποίους δημιουργήθηκε μία διακοπή. Ο επεξεργαστής με την σειρά του συνδέει τα σήματα διακοπών με συγκεκριμένες ρουτίνες λογισμικού οι οποίες είναι υπεύθυνες να εξυπηρετήσουν την διακοπή. Αυτές οι ρουτίνες ονομάζονται Ρουτίνες Χειρισμού Διακοπών (Interrupt Handling Routines) και στις περισσότερες περιπτώσεις η δουλειά τους είναι να διαβάσουν τον καταχωρητή διακοπών και ανάλογα με την αιτία διακοπής να καλέσουν άλλες ρουτίνες υψηλότερου επιπέδου οι οποίες θα αναλάβουν την αλληλεπίδραση με τον επεξεργαστή ώστε να εξυπηρετήσουν την διακοπή, δηλαδή να παρέχουν ή να δεχθούν πληροφορία από το στοιχείο. Κάθε φορά που ενεργοποιείται μία διακοπή προς τον επεξεργαστή αυτός είναι υποχρεωμένος να σταματήσει όλες τις τρέχουσες λειτουργίες του έως ότου ολοκληρωθεί η εκτέλεση της Ρουτίνας Χειρισμού Διακοπών.

Είναι προφανές ότι μέσω των διακοπών ο επεξεργαστής δεν είναι ανάγκη να εκτελεί άχρηστους βρόγχους ελέγχου των στοιχείων οι οποίοι κλέβουν χρόνο εκτέλεσης από άλλες κρίσιμες λειτουργίες του επεξεργαστή, ενώ επίσης η απόκριση του επεξεργαστή σε εξωτερικά γεγονότα γίνεται άμεσα χωρίς καθυστέρηση. Από την άλλη όμως οι διακοπές απαιτούν επιπλέον λογισμικό χειρισμού ενώ ταυτόχρονα στερούν από τον επεξεργαστή την ευχέρια να οργανώνει μόνος του τις διάφορες διαδικασίες. Έτσι σε περιπτώσεις που η εξυπηρέτηση ενός στοιχείου δεν είναι κρίσιμης χρονικά σημασίας μία διακοπή θα αναγκάσει τον επεξεργαστή να αναβάλει άλλες λειτουργίες του υψηλότερου επιπέδου οι οποίες είναι περισσότερο κρίσιμες.

Συμπερασματικά μπορεί να πει κανείς ότι για διαδικασίες αλληλεπίδρασης οι οποίες είναι κρίσιμης χρονικής σημασίας και μη προβλέψιμες ως προς τον χρόνο ενεργοποίησης τους είναι απαραίτητο να χρησιμοποιούνται οι διακοπές ως μέσο συγχρονισμού ενώ για διαδικασίες αλληλεπίδρασης οι οποίες δεν είναι κρίσιμης χρονικής σημασίας και προβλέψιμες (πιθανόν περιοδικές) είναι λογικό να χρησιμοποιείται το rolling από το λογισμικό.

Στην κάρτα του Εξυπηρετητή ABR χρησιμοποιήθηκαν και οι δύο αυτές τεχνικές. Τα στοιχεία της κάρτας εκτός της M.E.K.A. υποστηρίζουν διακοπές και συνδέονται με κατάλληλα pins του επεξεργαστή. Τα pins του επεξεργαστή MPC 860 που συνδέονται στα σήματα διακοπών των διαφόρων στοιχείων φαίνονται στον Πίνακα 5

<i>MPC860 pin</i>	<i>Σήμα Διακοπής</i>	<i>Φυσικό Στοιχείο</i>
IRQ0	IRQ0	-
IRQ1	IRQ1	-
RSV/IRQ 2	IRQ2	-
CR/IRQ3	IRQ3	CUBIT 1
KR/IRQ4	IRQ4	CUBIT2
DP2/IRQ5	PHY_INTB	PM5250
FRZ/IRQ6	IRQ6	MC92501
IRQ7	IRQ7	FPGA

**Πίνακας 5 Πίνακας Διακοπών του επεξεργαστή**

#### **3.2.1.4 Οντότητες Λογισμικού Χαμηλού Επιπέδου**

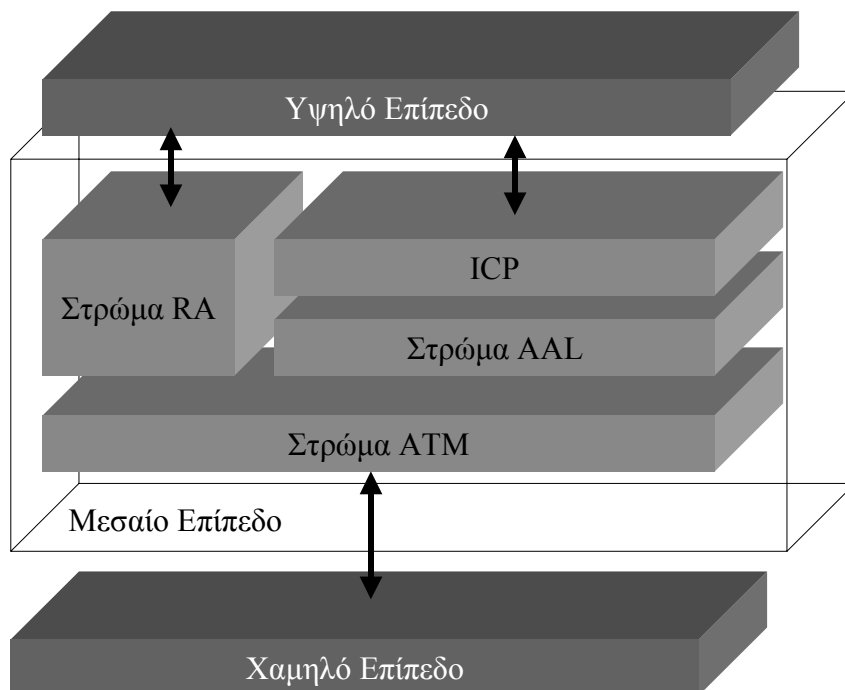
Σύμφωνα με τα παραπάνω το λογισμικό χαμηλού επιπέδου περιέχει:

- ✓ οντότητες οι οποίες εγγράφουν ή αναγνώσκουν από καθορισμένες αντιστοιχισμένες διευθύνσεις τιμές που περιέχονται ή προορίζονται στα στοιχεία της κάρτας
- ✓ οντότητες που εκτελούν βρόχους rolling για την αλληλεπίδραση με τα στοιχεία της κάρτας
- ✓ οντότητες που υλοποιούν ρουτίνες χειρισμού διακοπών (ISRs) για την αλληλεπίδραση με τα στοιχεία της κάρτας.

#### **3.2.2 Μεσαίο Επίπεδο**

Το Μεσαίο Επίπεδο του λογισμικού της κάρτας Εξυπηρετητή ABR αποτελείται από το σύνολο των πρωτοκόλλων επικοινωνίας. Τα πρωτόκολλα αυτά υλοποιούν τις

βασικές λειτουργίες των χαμηλών επιπέδων του προτύπου ATM έτσι όπως έχει οριστεί από το ATM Forum, και παρέχουν το απαραίτητο υπόβαθρο επικοινωνίας για τις εφαρμογές του Υψηλού Επιπέδου (Διαχείριση, Αλγόριθμος Ελέγχου Ροής). Οι λειτουργίες που υλοποιούνται στο Μεσαίο Επίπεδο καθώς και η σύνδεση τους με τις εφαρμογές του Υψηλού Επιπέδου φαίνονται στην Εικόνα 24



Εικόνα 24 Δομή του Μεσαίου Επιπέδου

### 3.2.2.1 Στρώμα ATM

Αυτό το στρώμα χρησιμοποιεί λειτουργίες του χαμηλού επιπέδου προκειμένου να διαβάσει ή να γράψει ένα cell από ή στον Επεξεργαστή Κελιών. Στη συνέχεια αναλαμβάνει να αναγνωρίσει τον τύπο του cell (Διαχείρισης ή RM) και να το αποθηκεύσει στην μνήμη του επεξεργαστή. Η αποθήκευση του cell γίνεται σε ειδικές δομές που δημιουργεί το λογισμικό και οι οποίες είναι αναγνωρίσιμες από τα υψηλότερα επίπεδα. Επίσης το συγκεκριμένο στρώμα εκτελεί λειτουργίες τοποθέτησης σε ουρές εισόδου των cells που λαμβάνει. Για κάθε στρώμα του αμέσως υψηλότερου επιπέδου διατηρείται και μία ξεχωριστή ουρά και τα cells τοποθετούνται ανάλογα με την αναγνώριση του τύπου τους. Αυτές οι ουρές είναι αναγνώσιμες από τα υψηλότερα επίπεδα. Στην αντίθετη κατεύθυνση το στρώμα ATM λαμβάνει cells από τα υψηλότερα επίπεδα λογισμικού με τις ίδιες αναγνωρίσιμες δομές, τα τοποθετεί σε ουρές εξόδου και αναλαμβάνει την εγγραφή τους στον Επεξεργαστή Κελιών, καλώντας λειτουργίες του Χαμηλού Επιπέδου.

### 3.2.2.2 Στρώμα AAL

Αυτό το στρώμα λαμβάνει cells Διαχείρισης (που προέρχονται από την κάρτα Κεντρικού Επεξεργαστή) από το στρώμα ATM και αναλαμβάνει να δημιουργήσει

πακέτα Διαχείρισης τα οποία προορίζονται για το πρωτόκολλο Διαχείρισης του υψηλότερου επιπέδου (ICP). Στην αντίθετη κατεύθυνση δέχεται πακέτα διαχείρισης από το ICP και τα μετατρέπει σε cells προκειμένου να μεταφερθούν μέσω του στρώματος ATM στον Επεξεργαστή Κελιών και κατ'έκταση στην κάρτα Κεντρικού Επεξεργαστή. Το συγκεκριμένο στρώμα υλοποιεί το πρωτόκολλο AAL5 και τη λειτουργία Segmentation and Reassembly (SAR) όπως αυτά ορίζονται από τα πρότυπα του ATM Forum[6].

### **3.2.2.3 Εσωτερικό Πρωτόκολλο Επικοινωνίας / Internal Communication Protocol(ICP)**

Στο στρώμα αυτό υλοποιείται η τοπική έκδοση του ICP. Η περιγραφή του πρωτοκόλλου βρίσκεται αναλυτικά στην παράγραφο 1.6.

Η τοπική έκδοση υλοποιεί ένα client ICP. Δέχεται ICP πακέτα μέσω του στρώματος AAL τα οποία επεξεργάζεται και εκτελεί τις κατάλληλες λειτουργίες. Τα πακέτα ICP που δέχεται η τοπική έκδοση του ICP είναι τα PING, PING\_REPLAY, OPEN\_CONNECTION, CLOSE\_CONNECTION.

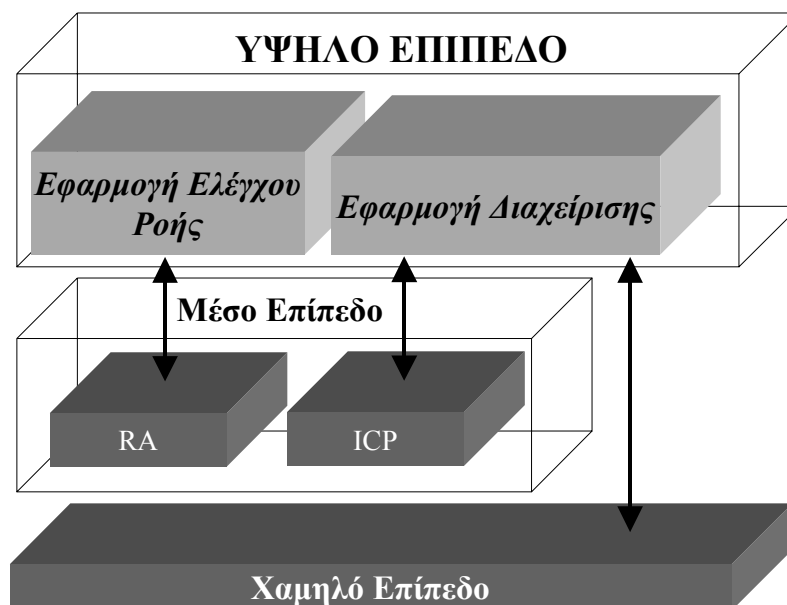
Επίσης επικοινωνεί με το επίπεδο διαχείρισης όταν πρόκειται για ICP πακέτα διαχείρισης και το ενημερώνει για νέες συνδέσεις. Τέλος αποστέλει και reply πακέτα γραφοντάς τα στον οδηγό AAL.

### **3.2.2.4 Στρώμα RA (RM cell Adaption Layer)**

Αυτό το στρώμα λαμβάνει RM cells μέσω του στρώματος ATM και αναλαμβάνει να εξάγει από τα πεδία τους τις κατάλληλες πληροφορίες οι οποίες είναι απαραίτητες για τον αλγόριθμο ελέγχου ροής του Υψηλού Επιπέδου. Αυτές οι πληροφορίες τοποθετούνται σε κατάλληλες δομές και μεταφέρονται στο Υψηλό Επίπεδο προς επεξεργασία και ανανέωση. Στην αντίθετη κατεύθυνση, το συγκεκριμένο στρώμα λαμβάνει από το Υψηλό Επίπεδο τις επεξεργασμένες και ανανεωμένες πληροφορίες και τις τοποθετεί στα κατάλληλα πεδία των cells τα οποία στην συνέχεια επιστρέφει πίσω στο στρώμα ATM για αποστολή προς τον Επεξεργαστή Κελιών.

## **3.2.3 Υψηλό Επίπεδο**

Στο Υψηλό Επίπεδο περιέχονται τα προγράμματα που υλοποιούν τις κυρίως λειτουργίες της Κάρτας Εξυπηρετητή ABR. Πιο συγκεκριμένα περιέχονται οι εφαρμογές Διαχείρισης και Ελέγχου Ροής. Η δομή του Υψηλού Επιπέδου καθώς και η σύνδεση του με τα χαμηλότερα επίπεδα φαίνονται στην Εικόνα 25



Εικόνα 25 Η δομή του Υψηλού Επιπέδου και η σύνδεση του με τα χαμηλότερα επίπεδα

### 3.2.3.1 Εφαρμογή Διαχείρισης

Η εφαρμογή Διαχείρισης λαμβάνει πακέτα διαχείρισης που προέρχονται από τον Κεντρικό Επεξεργαστή αφού αυτά υποστούν την επεξεργασία του στρώματος ICP. Όταν το client ICP που τρέχει στον τοπικό επεξεργαστή δεχθεί ένα μήνυμα *ICP OPEN CONNECTION* ή *ICP CLOSE CONNECTION* αφαιρεί τους ICP headers και προωθεί το μήνυμα στην εφαρμογή Διαχείρισης (βλ. Παρ.1.6.3.1). Η εφαρμογή Διαχείρισης αφού λάβει ένα πακέτο Διαχείρισης το αναλύει και τοποθετεί τα διάφορα πεδία πληροφορίας σε κατάλληλες δομές που διατηρεί στην τοπική μνήμη. Στην συνέχεια αναλύει τον τύπο του μηνύματος και λαμβάνει κατάλληλες αποφάσεις σχετικά με τις δράσεις που πρέπει να αναλάβει ώστε να μεταφερθεί η πληροφορία που προέρχεται από τον Κεντρικό Επεξεργαστή, στα στοιχεία της κάρτας. Πιο συγκεκριμένα η Εφαρμογή Διαχείρισης ενεργοποιεί απευθείας λειτουργίες του Χαμηλού Επιπέδου προκειμένου να ενημερώσει τα στοιχεία της κάρτας (Μ.Ε.Κ.Α. και Επεξεργαστή Κελιών) για τις αλλαγές στην κατάσταση των συνδέσεων, για της παραμέτρους των συνδέσεων καθώς και να εγγράψει πληροφορία στο εσωτερικό των στοιχείων της κάρτας καθώς και στις μνήμες τους.

Η εφαρμογή Διαχείρισης εκτός από τις λειτουργίες ενημέρωσης της κάρτας και εξυπηρέτησης των αιτήσεων του Κεντρικού Επεξεργαστή, εκτελεί και λειτουργίες τοπικού ελέγχου. Για αυτόν τον σκοπό εκτελεί περιοδικούς ελέγχους στα στοιχεία της κάρτας σε δεδομένα χρονικά διαστήματα, χρησιμοποιώντας λειτουργίες του Χαμηλού Επιπέδου για την προσπέλαση σε εσωτερικές δομές των στοιχείων οι οποίες περιέχουν παραμέτρους χρήσιμες για τον έλεγχο της ορθής λειτουργίας και της κατάστασης τους.



### 3.2.3.2 Εφαρμογή Ελέγχου Ροής

Η εφαρμογή ελέγχου ροής αποτελεί την υλοποίηση του αλγορίθμου Explicit Rate Flow Control. Αυτή η εφαρμογή τρέχει πάνω από το επίπεδο RA. Δέχεται από αυτό το επίπεδο της παραμέτρους που απαιτούνται για τον έλεγχο ροής και οι οποίες έχουν εξαχθεί από τα RM cells στο επίπεδο RA. Με αυτές τις παραμέτρους εκτελεί έναν αλγόριθμο Explicit Rate Flow Control και υπολογίζει την ανανεωμένες παραμέτρους λαμβάνοντας υπόψη και την τρέχουσα κατάσταση των πόρων του συστήματος βασιζόμενος σε πληροφορία που ανταλλάσσεται μέσω του πρωτοκόλλου ICP με τον Κεντρικό Επεξεργαστή του μεταγωγέα.

Αφού υπολογιστούν οι νέες παράμετροι εισάγονται στα RM cells στα οποία αντιστοιχούν μέσω του επιπέδου RA και κατόπιν μέσω των λειτουργιών του χαμηλού επιπέδου εισάγονται στον Επεξεργαστή Κελιών και απο εκεί στην κανονική ροή των cells κατευθυνόμενα προς την έξοδο του μεταγωγέα.

Στη συγκεκριμένη εφαρμογή επιλέχθηκε ένας συγκεκριμένος αλγόριθμος από ένα σύνολο πολλών και διαφορετικού χαρακτήρα αλγορίθμων. Ο αλγόριθμος που επιλέχθηκε να υλοποιηθεί στην κάρτα Εξυπηρετητή ABR ονομάζεται UCSC Scheme και αποτελεί μετασχηματισμό του βασικού αλγορίθμου ελέγχου ροής MIT [11][12].

Ο συγκεκριμένος αλγόριθμος επιλέχθηκε για τους παρακάτω λόγους:

- Είναι απλός στην υλοποίηση, απλότητα που χαρακτηρίζει όλους τους αλγορίθμους MIT
- Αποτελεί βελτιστοποίηση του αλγορίθμου MIT ως προς την υπολογιστική πολυπλοκότητα ( $O(1)$  σε σχέση με  $O(N)$  του MIT) το οποίο αποτελεί βασικό στοιχείο για μία κάρτα που μπορεί να υποστηρίξει έως και 32000 συνδέσεις. Το κόστος για την μείωση της πολυπλοκότητας είναι η αύξηση του χρόνου σύγκλισης και του απαιτούμενου αποθηκευτικού χώρου το οποίο είναι αποδεκτό με βάση τους πόρους του συστήματος
- Η πληροφορία (παράμετροι κατάστασης) που απαιτείται για την εκτέλεση του αλγορίθμου δεν χρειάζεται να ανακτηθεί μέσω πολύπλοκων διαδικασιών καθώς η ίδια πληροφορία βρίσκεται ήδη στην υπηρεσία του λογισμικού και για άλλες λειτουργίες του (Διαχείριση Συνδέσεων). Η ανάκτηση αυτής της πληροφορίας γίνεται ούτως ή άλλως μέσω της επικοινωνίας με της κεντρικές οντότητες διαχείρισης του μεταγωγού κάθε φορά που εγκαθίσταται ή καταργείται μία σύνδεση.

Η λειτουργία του αλγορίθμου ελέγχου ροής UCSC περιγράφεται γενικά στο Παράρτημα Β (B.2).

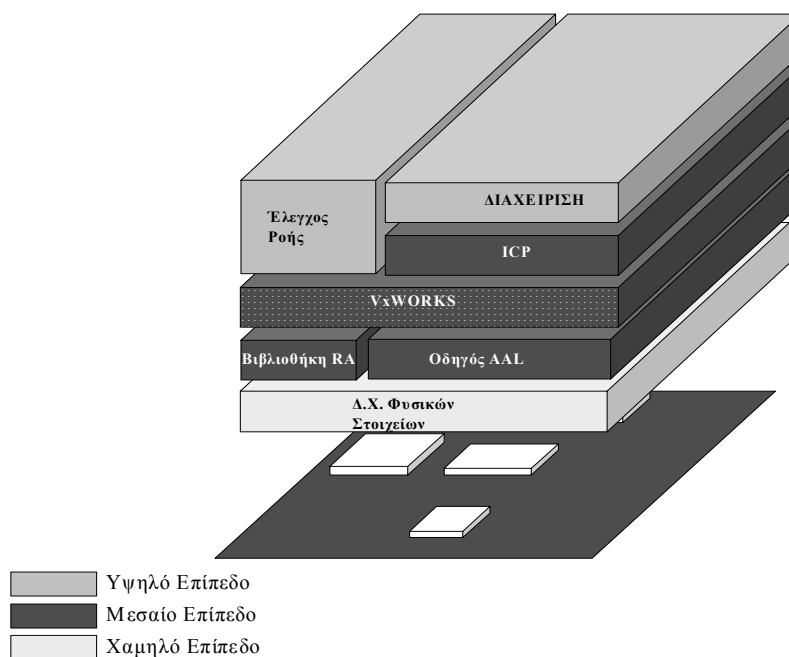
## 3.3 Σχεδίαση Λογισμικού

Η σχεδίαση του λογισμικού της κάρτας Εξυπηρετητή ABR βασίζεται στην ιεραρχική μορφή που φαίνεται στην Εικόνα 26. Το σύνολο του λογισμικού χτίστηκε γύρω από την πλατφόρμα που παρέχει το Λειτουργικό Σύστημα πραγματικού χρόνου VxWorks

της εταιρίας WindRiver Systems[10]. Οι δυνατότητες που παρέχει το λειτουργικό και πάνω στις οποίες δομήθηκε το υπόλοιπο λογισμικό περιγράφονται στην παράγραφο 3.3.1.

Στην Εικόνα 26 φαίνονται επίσης αναλυτικά οι οντότητες που συνθέτουν το λογισμικό της κάρτας εξυπηρετητή ABR. Η πρώτη κατηγορία οντοτήτων περιέχει το κομμάτι του λογισμικού που εκτελεί τις λειτουργίες του Χαμηλού Επιπέδου και αλληλεπιδρά απευθείας με το υλικό, δηλαδή με τα στοιχεία της κάρτας. Αυτή η κατηγορία σχεδιάστηκε με την μορφή Διεπαφών Χειρισμού Φυσικών Στοιχείων-ΔΧΦΣ η οποία είναι στην ουσία ένα API (Application Programming Interface) χαμηλού επιπέδου και παρέχει στις υπολοιπες οντότητες την απαραίτητη διεπαφή με το υλικό πάνω στην οποία θα σχεδιαστούν με την απαραίτητη ανεξαρτησία από αυτό. Η δεύτερη κατηγορία οντοτήτων περιέχει έναν οδηγό(driver) και μία βιβλιοθήκη(library) ειδικού σκοπού τα οποία σχεδιάστηκαν ειδικά για την κάρτα Εξυπηρετητή ABR και τα οποία προορίζονται για ενσωμάτωση στον πυρήνα του λειτουργικού συστήματος μαζί με τις υπόλοιπες παρομοιες οντότητες (οδηγοί βιβλιοθήκες) γενικού σκοπού που υπάρχουν σε αυτό. Πιο συγκεκριμένα ο οδηγός που σχεδιάστηκε είναι ένας Οδηγός AAL5 ο οποίος εκτελεί τις λειτουργίες του ομώνυμου στρώματος του ATM, ενώ η βιβλιοθήκη περιέχει ειδικές συναρτήσεις για τον χειρισμό των πεδίων των RM cells και υλοποιεί το επίπεδο RA όπως περιγράφεται στην παράγραφο 3.2.2.4. Τέλος η τελευταία κατηγορία οντοτήτων περιέχει όλα τα προγράμματα των δύο υψηλότερων επιπέδων τα οποία εκτελούνται στον επεξεργαστή και υλοποιούν όλες τις απαραίτητες λειτουργίες. Οι οντότητες που περιέχονται σε αυτή την κατηγορία είναι αυτές που περιγράφονται στο κεφάλαιο 3.2:

- Εσωτερικό Πρωτόκολλο Επικοινωνίας (ICP)
- Διαχείριση
- Αλγόριθμος Ελέγχου Ροής



Εικόνα 26 Επίπεδα Σχεδίασης Λογισμικού

### 3.3.1 Το Λειτουργικό Σύστημα (VxWorks)

Η σχεδίαση και υλοποίηση του λογισμικού της κάρτας Εξυπηρετητή ABR βασίστηκε στην πλατφόρμα του λειτουργικού συστήματος **VxWorks** της εταιρίας *WindRiver Systems* [10]. Αυτό το λειτουργικό σύστημα πραγματικού χρόνου (real time OS) το οποίο είναι ρυθμισμένο για να τρέχει στον επεξεργαστή της κάρτας (MPC860SAR της Motorola) έχει τα εξής χαρακτηριστικά που χρησιμοποιήθηκαν κατά την ανάπτυξη και υλοποίηση:

- **Υποστήριξη πολλών ταυτόχρονων διεργασιών (Multitasking) και επικοινωνία μεταξύ διεργασιών (Intertask Communication):** Τα μοντέρνα συστήματα πραγματικού χρόνου βασίζονται στις συμπληρωματικές λειτουργίες Multitasking και Intertask Communication. Ένα περιβάλλον multitasking επιτρέπει στις εφαρμογές πραγματικού χρόνου να υλοποιηθούν σαν ένα σύνολο ανεξάρτητων διεργασιών (tasks), κάθε μία από τις οποίες εκτελείται σε ξεχωριστό thread και χρησιμοποιεί το δικό της σύνολο πόρων συστήματος. Οι λειτουργίες επικοινωνίας μεταξύ των διεργασιών (Intertask Communication) επιτρέπουν στις διεργασίες να συγχρονίσουν και παραλληλίσουν την εκτέλεση τους.

Ο multitasking πυρήνας του VxWorks, χρησιμοποιεί χρονοπρογραμματισμό διεργασιών είτε οδηγούμενο από διακοπές (interrupt driven) είτε βασισμένο σε προτεραιότητες (priority-based). Χαρακτηρίζεται από γρήγορους χρόνους context switch και από χαμηλή καθυστέρηση απόκρισης σε διακοπές. Στο VxWorks κάθε υπορουτίνα μπορεί να *εκκινηθεί (spawned)* ως μία ξεχωριστή διεργασία, με το δικό της context και τη δική της στοίβα. Άλλες βασικές λειτουργίες ελέγχου διεργασιών επιτρέπουν σε αυτές να μούν σε *αναμονή (suspended)* να

επανεκκινηθούν (*resume*) να διαγραφούν (*delete*), να καθυστερήσουν (*delay*) και να αλλάξουν επίπεδο προτεραιότητας (*move in priority*).

Ο πυρήνας του VxWorks παρέχει σημαφόρους (semaphores) ως τον βασικό μηχανισμό για τον συγχρονισμό και τον αμοιβαίο αποκλεισμό των διεργασιών. Οι σημαφόροι αυτοί είναι γρήγοροι στην εκτέλεση τους και απλοί στην χρήση τους.

Για την επικοινωνία μεταξύ διεργασιών, ο πυρήνας του VxWorks παρέχει επίσης message queues, pipes, sockets και signals.

- **Σύστημα Εισόδου/Εξόδου (I/O System):** Το σύστημα E/E του VxWorks παρέχει ομοιόμορφη και ανεξάρτητη από το είδος στοιχείου, πρόσβαση σε περιφερειακά υποσυστήματα. Ο χρήστης μπορεί να καλέσει επτά βασικές ρουτίνες E/E: *creat()*, *remove()*, *open()*, *close()*, *read()*, *write()* και *ioctl()*. Επίσης παρέχονται υψηλού επιπέδου ρουτίνες E/E όπως *printf()* και *scanf()*.

Το VxWorks περιέχει οδηγούς περιφερειακών συσκευών για σειριακή επικοινωνία, δίσκους, επικοινωνία μεταξύ διεργασιών (pipes) και συσκευές δικτύου. Οι σχεδιαστές εφαρμογών μπορούν εύκολα να γράψουν ξεχωριστούς ειδικευμένους οδηγούς αν είναι αναγκαίο. Το VxWorks επίσης επιτρέπει την δυναμική εγκατάσταση και αφαίρεση οδηγών χωρίς να χρειάζεται η επανεκκίνηση του συστήματος. Τέλος επιτρέπει στους επιμέρους οδηγούς να έχουν τον συνολικό έλεγχο όσον αφορά τον τρόπο εξυπηρέτησης των αιτήσεων των χρηστών. Έτσι οι οδηγοί μπορούν εύκολα να υλοποιήσουν διαφορετικά πρωτόκολλα και ειδικευμένες ρουτίνες χειρισμού συσκευών χωρίς να παρεμβολή από το ίδιο το σύστημα E/E.

- **Εργαλεία ανάπτυξης ενσωματωμένα στο σύστημα:** Το VxWorks συνοδεύεται από μία σουίτα εργαλείων ανάπτυξης που εκτελούνται στο host μηχάνημα. Ωστόσο, τα εργαλεία ανάπτυξης του VxWorks μπορούν να χρησιμοποιηθούν και αυτόνομα στο target σύστημα, μέσω του κελύφους (shell), του πίνακα συμβόλων (symbol table) και του module loader/unloader που μπορούν να ενσωματωθούν στο λειτουργικό σύστημα. Αυτό εκτός των άλλων παρέχει ευελιξία στην ανάπτυξη εφαρμογών και την δυνατότητα δημιουργίας ενός δυναμικά ρυθμιζόμενου συστήματος πραγματικού χρόνου.
- **Βοηθητικές Βιβλιοθήκες (Utility Libraries):** Το VxWorks παρέχει πολλές γενικές βοηθητικές υπορουτίνες οι οποίες είναι οργανωμένες σε βιβλιοθήκες.
  - ✓ **Υποστήριξη χειρισμού διακοπών:** Το VxWorks παρέχει ρουτίνες για τον χειρισμό διακοπών που προέρχονται από το υλικό, χωρίς να χρειάζεται ο χρήστης να καταφύγει στην χρησιμοποίηση assembly. Παρέχονται ρουτίνες που συνδέουν άλλες ρουτίνες C του χρήστη σε διανύσματα διακοπών του υλικού (hardware interrupt vectors) και ρυθμίζουν τα επίπεδα των διακοπών του επεξεργαστή.
  - ✓ **Watchdog Timers:** Η λειτουργία watchdog επιτρέπει στους χρήστες της να προγραμματίσουν την εκτέλεση των ρουτίνων τους μετά από συγκεκριμένα χρονικά διαστήματα. Όταν περάσουν ένας προκαθορισμένος αριθμός

κομματιών χρόνου, μία επίσης προκαθορισμένη «timeout» ρουτίνα καλλείται στο επίπεδο διακοπής του ρολογιού συστήματος, εφόσον δεν έχει ακυρωθεί προηγουμένως ο watchdog.

- ✓ *Καταγραφή μηνυμάτων (message logging)*: Μία απλή λειτουργία καταγραφής μηνυμάτων επιτρέπει στις εφαρμογές να αποστείλουν μηνύματα σφαλμάτων ή κατάστασης σε μία διεργασία καταγραφής (logging task), η οποία στη συνέχεια σχηματίζει και παρέχει σε μία συσκευή καταγραφής (κονσόλα συστήματος, δίσκος, μνήμη) τα μηνύματα αυτά.
- ✓ *Δέσμευση μνήμης*: Το VxWorks παρέχει μία λειτουργία διαχείρισης μνήμης χρήσιμη για την δυναμική δέσμευση/αποδέσμευση και επαναδέσμευση blocks μνήμης. Blocks οποιουδήποτε μεγέθους μπορούν να δεσμευτούν και ο χρήστης μπορεί να καθορίσει το μέγεθός τους.
- ✓ *Σχηματισμός και ανάλυση αλφαριθμητικών (String Formatting and Scanning)*: Το VxWorks περιέχει ένα ολοκληρωμένο σύνολο υπορουτινών σχηματισμού και ανάλυσης αλφαριθμητικών της βιβλιοθήκης ANSI C, οι οποίες υλοποιούν τις printf()/scanf() και άλλες σχετικές ρουτίνες.
- ✓ *Βιβλιοθήκες ANSI C* : Το VxWorks παρέχει όλες τις βιβλιοθήκες C που προσδιορίζονται από το πρότυπο ANSI X3.159-1989. Οι πιο συχνά χρησιμοποιούμενες βιβλιοθήκες είναι:
  - ◆ **ansiCtype** ρουτίνες για τον χειρισμό χαρακτήρων
  - ◆ **ansiMath** τριγωνομετρικές, εκθετικές και λογαριθμικές ρουτίνες
  - ◆ **ansiSetjmp** ρουτίνες για την υλοποίηση μη τοπικού goto
  - ◆ **ansiStdarg** ρουτίνες για την χρήση λίστας παραμέτρων μεταβλητού μεγέθους
  - ◆ **ansiStdio** ρουτίνες για τον χειρισμό ροών E/E
  - ◆ **ansiStdlib** σύνολο ρουτινών για μετάφραση τύπων μεταβλητών, δέσμευση μνήμης και γέννεση τυχαίων αριθμών
  - ◆ **sigLib** ρουτίνες χειρισμού signals
- *Μέτρηση και ανάλυση απόδοσης*: Για την κατανόηση και βελτιστοποίηση της απόδοσης ενός συστήματος πραγματικού χρόνου, είναι χρήσιμη η χρονομέτρηση των ρουτινών που χρησιμοποιούνται. Για αυτόν τον λόγο το VxWorks παρέχει χρήσιμα εργαλεία χρονομέτρησης, όπως οι execution timers.

Οι execution timers του VxWorks μπορούν να μετρήσουν τον χρόνο εκτέλεσης οποιασδήποτε υπορουτίνας ή ομάδας υπορουτινών. Επειδή το ρολόι συστήματος είναι πολύ αργό για να παρέχει την αναγκαία ανάλυση για την χρονομέτρηση πολύ γρήγορων ρουτινών, οι timers έχουν την δυνατότητα να εκτελούν και να

μετρούν επαναληπτικά ένα σύνολο ρουτινών έως ότου η μέτρηση επιτύχει μία ικανοποιητική ακρίβεια.

- **Board Support Packages (BSPs):** Το VxWorks περιέχει βιβλιοθήκες που το καθιστούν πλήρως μεταφέρσιμο (portable). Αυτές παρέχουν μία όμοια διεπαφή λογισμικού για τις λειτουργίες του υλικού σε πολλά γνωστά boards. Περιέχουν λειτουργίες αρχικοποίησης του υλικού, χειρισμού και δημιουργίας διακοπών, διαχείριση του ρολογιού και των timers του υλικού, αντιστοίχιση των τοπικών ή των εξωτερικών χώρων διευθύνσεων (memory map) κ.λ.π.

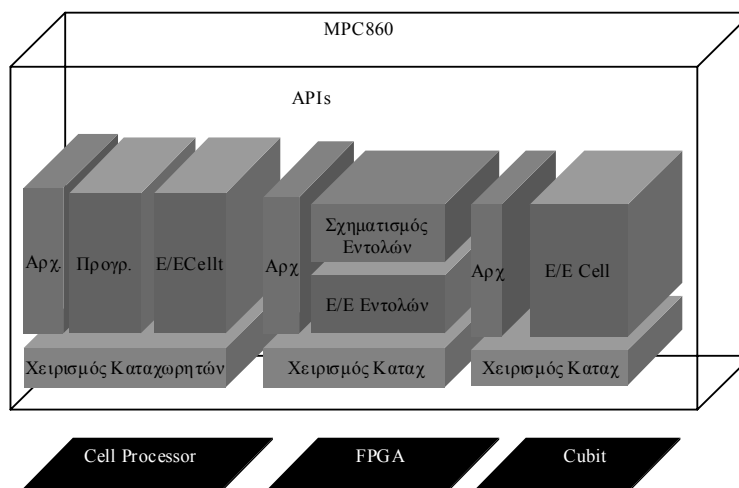
### 3.3.2 Διεπαφές Χειρισμού Φυσικών Στοιχείων (Δ.Χ.Φ.Σ.)

Το λογισμικό θα βασιστεί σε τρία ΔΧΦΣ, ένα για κάθε φυσικό στοιχείο της κάρτας. Τα ΔΧΦΣ περιέχουν τα εξής στοιχεία:

- Περιγραφή σε λογισμικό των διεπαφών επεξεργαστή των τριών στοιχείων, χρησιμοποιώντας δείκτες σε αντιστοιχισμένες θέσεις μνήμης που αντιστοιχούν σε εσωτερικούς επεξεργαστές των στοιχείων.
- Συναρτήσεις αρχικοποίησης και προγραμματισμού των στοιχείων.
  - Εγγραφή-ανάγνωση εσωτερικών καταχωρητών αρχικοποίησης (Configuration Registers)
  - Εγγραφή-ανάγνωση εσωτερικών καταχωρητών προγραμματισμού (Control Registers)
  - Εγγραφή – Ανάγνωση εξωτερικής μνήμης των στοιχείων της κάρτας.
- Συναρτήσεις ανταλλαγής πληροφορίας μεταξύ των στοιχείων
  - Παροχή εντολών
  - Ανάγνωση αποτελεσμάτων
  - Εισαγωγή/Εξαγωγή cells
- Συναρτήσεις ελέγχου της κατάστασης των στοιχείων
- Ρουτίνες Εξυπηρέτησης Διακοπών (Interrupt Service Routines)

Εκτός από τις Ρουτίνες Εξυπηρέτησης Διακοπών, οι υπόλοιπες λειτουργίες των ΔΧΦΣ δεν εκτελούνται αυτόματα ως υποπρογράμματα αλλά χρησιμοποιούνται (καλούνται) από τα προγράμματα εφαρμογών που τρέχουν στα υψηλότερα επίπεδα αφαίρεσης.

Τα ΔΧΦΣ έχουν σχεδιαστεί και αυτά σε επίπεδα. Η δομή και λειτουργικότητα που παρέχουν τα ΔΧΦΣ καθώς και η αντιστοίχσή τους με τα στοιχεία της κάρτας φαίνεται στην Εικόνα 27.



Εικόνα 27 Δομή και αντιστοίχιση των ΔΧΦΣ

Η περιγραφή των μερών των ΔΧΦΣ περιγράφεται παρακάτω:

### 3.3.2.1 ΔΧΦΣ Επεξεργαστή Κελιών(ΔΧΦΣ/ΕΚ)

- **Χειρισμός Καταχωρητών:** Ορισμός της εσωτερικής αντιστοίχισης διευθύνσεων (memory map) και των format των καταχωρητών που συγκροτούν την λογική διεπαφή του Επεξεργαστή Κελιών με τον επεξεργαστή.
- **Αρχικοποίηση (Αρχ):** Ορισμός των τιμών των καταχωρητών για την αρχικοποίηση και τον προγραμματισμό του Επεξεργαστή Κελιών. Αυτό το επίπεδο χρησιμοποιείται μόνο κατά την εκκίνηση του συστήματος.
- **E/E (Εισαγωγή-Εξαγωγή) Cell:** Το σύνολο των συναρτήσεων που υλοποιούν το διάβασμα και γράψιμο cells από και προς τον Επεξεργαστή Κελιών μέσω των κατάλληλων διεπαφών του. Αυτό το στρώμα υλοποιεί την μεταφορά cells από το “Φυσικό επίπεδο” που είναι ο Επεξεργαστής Κελιών στον επεξεργαστή. Ο Επεξεργαστής Κελιών περιέχει μία ουρά καταχωρητών στην οποία αποθηκεύονται τα cells που κατευθύνονται προς τον επεξεργαστή και καταχωρητές για τα cells που προέρχονται από τον επεξεργαστή. Η δουλειά του επιπέδου E/E Cell είναι να διαβάζει και να γράφει αυτούς τους καταχωρητές και να αποθηκεύει προσωρινά σε έναν cell buffer τα cells εισόδου ή εξόδου.
- **Προγραμματισμός:** Περιέχει συναρτήσεις για τον προγραμματισμό του Επεξεργαστή Κελιών όσον αφορά την συμπεριφορά του στις διάφορες συνδέσεις και τους τύπους των cells που χειρίζεται. Πιο συγκεκριμένα προγραμματίζουν ειδικούς καταχωρητές του Ε.Κ. καθώς και εγγραφές στη μνήμη του ώστε να χειρίζεται με τον τρόπο που θέλουμε τα cells που δέχεται. Ο χειρισμός έχει να κάνει κυρίως με τους κανόνες εξαγωγής cells προς τον

Επεξεργαστή καθώς και με την επεξεργασία των cells που στέλνει ο Επεξεργαστής πρώτου αυτά μπουν στο κανονικό μονοπάτι δεδομένων του συστήματος.

Επίσης σε αυτό το στρώμα υλοποιείται και η Ρουτίνα Εξυπηρέτησης Διακοπών (Interrupt Service Routine) η οποία συνδέεται στο σήμα Διακοπής του Επεξεργαστή Κελιών και όταν υπάρξει διακοπή προσδιορίζει την αιτία της διαβάζοντας τον κατάλληλο Interrupt Register. Στη συνέχεια ενεργοποιεί μία ρουτίνα χειρισμού της διακοπής το οποίο υλοποιείται στο στρώμα AAL.

### 3.3.2.2 ΔΧΦΣ Στοιχείου Μεταγωγής (ΔΧΦΣ/ΣΜ)

- **Χειρισμός Καταχωρητών:** Ορισμός της εσωτερικής αντιστοίχισης διευθύνσεων (memory map) και των format των καταχωρητών που συγκροτούν την λογική διεπαφή των φυσικών στοιχείων με τον επεξεργαστή.
- **Αρχικοποίηση (Αρχ):** Ορισμός των τιμών των καταχωρητών για την αρχικοποίηση και τον προγραμματισμό του Στοιχείου Μεταγωγής. Αυτό το επίπεδο χρησιμοποιείται μόνο κατά την εκκίνηση του συστήματος.
- **E/E Cell:** Το σύνολο των συναρτήσεων που υλοποιούν το διάβασμα και γράψιμο cells από και προς το Στοιχείου Μεταγωγής μέσω των κατάλληλων διεπαφών του. Αυτό το στρώμα υλοποιεί την μεταφορά cells από το “Φυσικό επίπεδο” που είναι το Στοιχείου Μεταγωγής στον επεξεργαστή. Το Στοιχείου Μεταγωγής περιέχει μία ουρά καταχωρητών στην οποία αποθηκεύονται τα cells που κατευθύνονται προς τον επεξεργαστή και καταχωρητές για τα cells που προέρχονται από τον επεξεργαστή. Η δουλειά του επιπέδου E/E Cell είναι να διαβάζει και να γράφει αυτούς τους καταχωρητές και να αποθηκεύει προσωρινά σε έναν cell buffer τα cells εισόδου ή εξόδου.

Επίσης σε αυτό το στρώμα υλοποιείται και η Ρουτίνα Εξυπηρέτησης Διακοπών (Interrupt Service Routine) η οποία συνδέεται στο σήμα Διακοπής του Επεξεργαστή Κελιών και όταν υπάρξει διακοπή προσδιορίζει την αιτία της διαβάζοντας τον κατάλληλο Interrupt Register. Στη συνέχεια ενεργοποιεί μία ρουτίνα χειρισμού της διακοπής το οποίο υλοποιείται στο στρώμα AAL.

### 3.3.2.3 ΔΧΦΣ Μ.Ε.Κ.Α.

- **Χειρισμός Καταχωρητών:** Ορισμός της εσωτερικής αντιστοίχισης διευθύνσεων (memory map) και των format των καταχωρητών που συγκροτούν την λογική διεπαφή των φυσικών στοιχείων με τον επεξεργαστή.
- **Αρχικοποίηση:** Ορισμός των τιμών των καταχωρητών για την αρχικοποίηση και τον προγραμματισμό των φυσικών στοιχείων της κάρτας. Αυτό το επίπεδο χρησιμοποιείται μόνο κατά την εκκίνηση του συστήματος.



- **E/E Εντολών:** Το σύνολο των συναρτήσεων που υλοποιούν το γράψιμο εντολών στην Μ.Ε.Κ.Α και το διάβασμα αποτελεσμάτων από αυτή μέσω των κατάλληλων καταχωρητών της διεπαφής της.
- **Σχηματισμός Εντολών:** Το σύνολο των συναρτήσεων που υλοποιούν την τοποθέτηση των παραμέτρων που περιέχουν οι εντολές σε κατάλληλα πεδία ώστε να επιτευχθεί η ορθή τοποθέτηση τους στους καταχωρητές της Μ.Ε.Κ.Α.

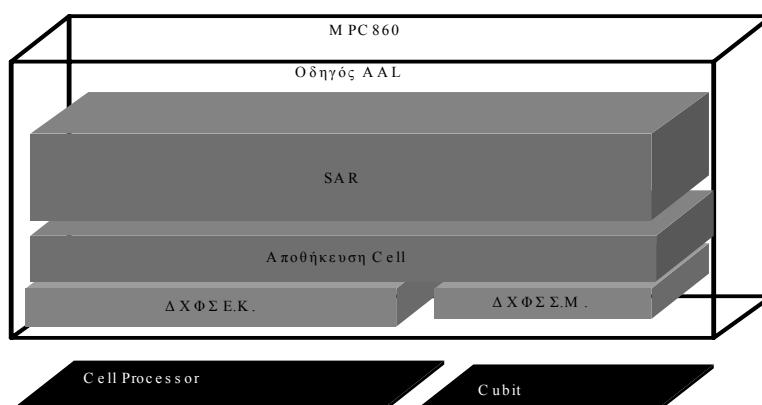
Αναλυτική περιγραφή των ΔΧΦΣ για κάθε στοιχείο καθώς και πλήρης πληροφορίες υλοποίησης περιέχονται στο επόμενο κεφάλαιο.

### 3.3.3 Οδηγός AAL

Ο επεξεργαστής χρησιμοποιεί τον AAL driver προκειμένου να δεχθεί τα πακέτα διαχείρισης που προέρχονται από τον κεντρικό επεξεργαστή και μεταφέρονται μέσω ειδικών cells. Επειδή αυτή η επικοινωνία γίνεται κατά κάποιον τρόπο μέσω ενός εσωτερικού ATM δικτύου, τόσο ο τοπικός όσο και ο κεντρικός επεξεργαστής θα πρέπει να υλοποιούν όλα τα απαραίτητα επίπεδα της στοίβας του ATM εκτός από το φυσικό επίπεδο [6]. Ο οδηγός AAL αναλαμβάνει την υλοποίηση αυτής της στοίβας.

Ο οδηγός AAL είναι παραμετροποιήσιμος όσον αφορά το φυσικό στοιχείο με το οποίο αλληλεπιδρά. Στην πραγματικότητα υλοποιήθηκαν δύο οδηγοί οι οποίοι διαφέρουν μόνο στο ότι χρησιμοποιούν διαφορετικό ΔΧΦΣ ανάλογα με το στοιχείο της κάρτας με το οποίο ο χρήστης της επιλέγει να υλοποιήσει την μεταφορά cells προς και από τον επεξεργαστή. Αυτά τα στοιχεία μπορεί να είναι είτε ο Επεξεργαστής Κελιών (MC92501) είτε το Στοιχείο Μεταγωγής (Cubit Pro). Κατά την εκκίνηση του συστήματος η οντότητα αρχικοποίησης καθορίζει ποιόν από τους δύο οδηγούς θα χρησιμοποιήσει.

Στην Εικόνα 28 φαίνεται η δομή του Οδηγού AAL καθώς και η προσαρμογή του στα διαφορετικά φυσικά στοιχεία



Εικόνα 28 Αναλυτική δομή του Οδηγού AAL

- **Αποθήκευση Cell:** Το σύνολο των συναρτήσεων που υλοποιούν την αποθήκευση των cells σε κατάλληλους buffers εσωτερικά του επεξεργαστή. Αυτό το επίπεδο υλοποιεί το στρώμα ATM στο εσωτερικό του επεξεργαστή και είναι κοινό σε όλους τους οδηγούς. Η λειτουργία αυτού του στρώματος είναι να εισάγει τα cells που προέρχονται από το φυσικό στοιχείο (MC92501 ή Cubit Pro) μέσω των ΔΧΦΣ και βρίσκονται αποθηκευμένα στην μνήμη σε έναν προσωρινό buffer ή τα cells που παράγονται από το επόμενο στρώμα (AAL) και κατευθύνονται προς τα φυσικά στοιχεία, σε κατάλληλες ουρές FIFO εισόδου ή εξόδου. Επίσης το στρώμα αυτό είναι υπεύθυνο και για την αντίστροφη διαδικασία, δηλαδή την εξαγωγή cells από τις FIFO ουρές και την αποστολή τους είτε στον προσωρινό buffer αποθήκευσης όταν κατευθύνονται προς το φυσικό στοιχείο (cells εξόδου) είτε προς τα ανώτερα στρώματα των οδηγών (cells εισόδου). Μέσω αυτού του επιπέδου πραγματοποιείται αποσύζευξη μεταξύ των λειτουργιών Cell Ins/Ext των ΔΧΦΣ των στοιχείων και των κυρίως επιπέδων των οδηγών (AAL, FC).
- **SAR:** Το σύνολο των συναρτήσεων που υλοποιούν την λειτουργία SAR για τα πακέτα διαχείρισης. Αυτό το επίπεδο εισάγει στο λειτουργικό τις συναρτήσεις διεπαφής του Οδηγού AAL (create(), open(), read(), write(), close()), ώστε να μπορούν να χρησιμοποιηθούν από τις εφαρμογές, καθώς και την συνάρτηση του Χειριστή Διακοπών (Interrupt Handler) για τον Οδηγό AAL.

Όπως όλοι οι οδηγοί του VxWorks έτσι και ο Οδηγός AAL μπορεί να εγκατασταθεί δυναμικά στο λειτουργικό σύστημα και να χρησιμοποιηθεί από τις εφαρμογές υψηλού επιπέδου μέσω γενικών εντολών οδηγών (open, close, read, write) χειριζόμενες περιγραφείς αρχείων (file descriptors)[10]. Μπορεί να εκτελείται είτε αυτόνομα ενεργοποιούμενες από διάφορα events, είτε ενεργοποιούμενες από τις εφαρμογές που τρέχουν από πάνω. Σε όλες τις περιπτώσεις η δουλειά του είναι να αλληλεπιδρά με το υλικό και να εκτελεί τα απαραίτητα πρωτόκολλα του Μεσαίου Επιπέδου (κυριώς τα στρώματα ATM και AAL) παρέχοντας τις κατάλληλες υπηρεσίες στις εφαρμογές του Υψηλού Επιπέδου.

Η λειτουργία των βασικών εντολών του οδηγού είναι:

- **AALcreate:** δημιουργεί τον οδηγό δίνοντας του ένα όνομα (device name) και σχετίζοντάς το φυσικό στοιχείο που χρησιμοποιεί. Επίσης δημιουργεί δομές ουρών ανάγνωσης και εγγραφής οι οποίες θα φιλοξενίσουν τα cells που επεξεργάζεται ο οδηγός χρησιμοποιώντας λειτουργίες του επιπέδου *Αποθήκευσης Cell*. Επίσης δημιουργεί δομές που περιέχουν τις αναγκαίες παραμέτρους για τον οδηγό όπως μετρητές μεγέθους πακέτων. Τέλος προγραμματίζει το φυσικό στοιχείο (MC92501 ή Cubit) ώστε να δέχεται και να μεταφέρει προς τον επεξεργαστή τα cells τα οποία προορίζονται για επεξεργασία από τον οδηγό.
- **AALOpen:** ξεκινά τον οδηγό παρέχοντας ένα διακριτικό του με την μορφή περιγραφέα αρχείου (file descriptor) με το οποίο οι εφαρμογές αναφέρονται στον οδηγό προκειμένου να τον χρησιμοποιήσουν.

- *AALClose*: κλείνει έναν οδηγό AAL και ελευθερώνει τον περιγραφέα αρχείου του.
- *AALRead*: διαβάζει ένα AAL πακέτο το οποίο δημιουργεί ο οδηγός από τα cells μίας ουράς ανάγνωσής του (Reassembly).
- *AALWrite*: γράφει ένα AAL πακέτο στον οδηγό ο οποίος το μετατρέπει σε cells τα οποία τοποθετούνται σε μία ουρά εγγραφής (Segmentation).

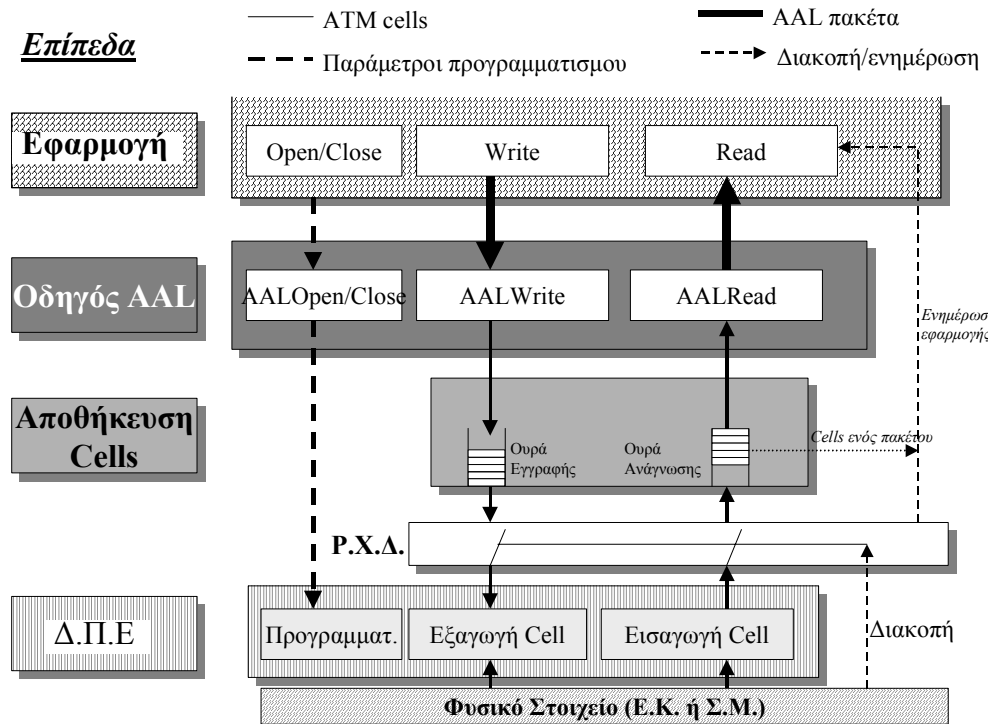
Στην Εικόνα 29 φαίνεται η λειτουργία του Οδηγού AAL ανά επίπεδο και η μεταφορά των δεδομένων που πραγματοποιείται.

Ο οδηγός AAL χειρίζεται cells τα οποία προέρχονται ή προορίζονται από/προς τα φυσικά στοιχεία της κάρτας και κατεπέκταση από/προς την φυσική ροή των cells. Αυτό συνεπάγεται ότι ο χειρισμός των cells εκ μέρους του επεξεργαστή θα πρέπει να προσαρμοστεί στην ασύγχρονη συμπεριφορά τους, για αυτό τον λόγο θα πρέπει να υλοποιείται με event driven τρόπο. Αυτό σημαίνει ότι η εφαρμογή που καλεί τον οδηγό δεν σε κάποιες περιπτώσεις θα τον καλεί σύγχρονα αποφασίζοντας η ίδια πότε θα συμβεί αυτό ενώ σε άλλες περιπτώσεις θα τον καλεί ασύγχρονα λαμβάνοντας υπόψη εξωτερικά γεγονότα. Για να υλοποιηθεί αυτή η λειτουργία, ενσωματώθηκε στον Οδηγό AAL μία *Ρουτίνα Εξυπηρέτησης Διακοπών (P.E.Δ.)*

Όπως φαίνεται και στην Εικόνα 29 στην περίπτωση της *AALRead* αυτή πρέπει να καλείται από την εφαρμογή κάθε φορά που πράγεται μία διακοπή από τα φυσικά στοιχεία της κάρτας που έχουν ένα νέο cell για τον επεξεργαστή. Αυτές οι διακοπές θα ενεργοποιούν την ειδική *P.E.Δ.* που βρίσκεται ενσωματωμένη στον οδηγό. Αυτή η ρουτίνα θα εκτελεί τις λειτουργίες ATM και θα ενημερώνει τις εφαρμογές ότι ο οδηγός είναι έτοιμος να τους παρέχει τις υπηρεσίες ανάγνωσής του. Οι εφαρμογές με τη σειρά τους μέσω της γενικής κλήσης του οδηγού (*AALRead*) η οποία υλοποιεί τη λειτουργία *AAL Reassembly*, θα διαβάζει πακέτα που μεταφέρονται μέσω των φυσικών στοιχείων.

Στην περίπτωση της *AALWrite* (Εικόνα 29) οι εφαρμογές μπορούν να την καλέσουν σύγχρονα (οποτεδήποτε θέλουν) και να γράψουν ένα πακέτο στην ουρά εξόδου του οδηγού. Ωστόσο η μεταφορά των cells από τις ουρές στα φυσικά στοιχεία είναι ασύγχρονη και ενεργοποιείται από την *P.E.Δ.* η οποία με τη σειρά της θα εκτελεί τις λειτουργίες ATM και χαμηλού επιπέδου.

Επιπρόσθετα οι ρουτίνες των γενικών κλήσεων (*AALRead*, *AALWrite*) εκτελούν και αυτές τις λειτουργίες ATM σε περίπτωση που τα φυσικά στοιχεία είναι σε θέση να δεχθούν ή να παρέχουν cells πρωτού τερματιστεί η εκτέλεση τους.



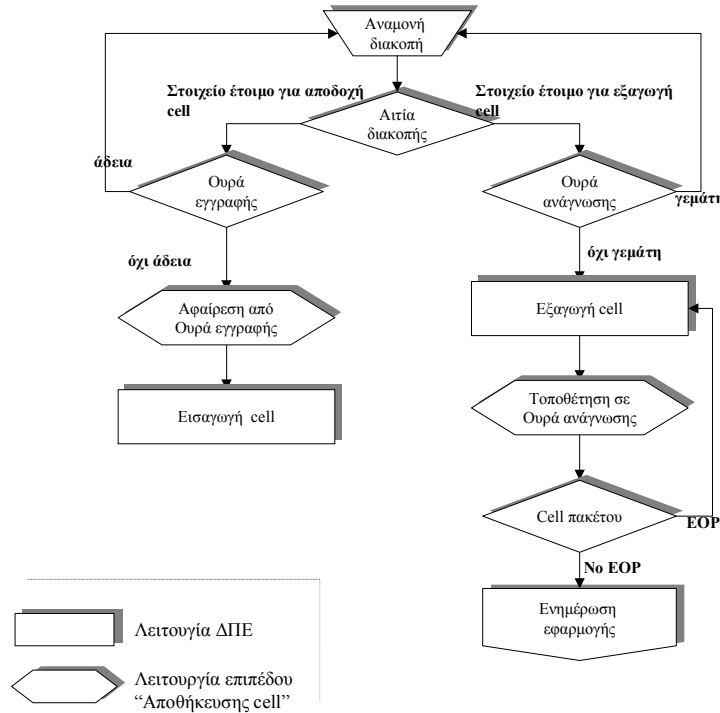
Εικόνα 29 Λειτουργία και μεταφορά δεδομένων στον Οδηγό AAL

Στον οδηγό AAL το πρώτο μέρος του στρώματος ATM υλοποιείται μέσω των συναρτήσεων που περιέχονται από το ΔΠΕ που θα χρησιμοποιηθεί (του Επεξεργαστή Κελιών ή του Cubit) και συγκεκριμένα από το στρώμα *E/E cell* (Εικόνα 27). Το δεύτερο μέρος του στρώματος ATM υλοποιείται από το επίπεδο *Αποθήκευσης Cell* (Εικόνα 28) το οποίο είναι κοινό για όλους τους οδηγούς. Το στρώμα AAL υλοποιείται από το ομώνυμο επίπεδο (Εικόνα 28).

Στην συνέχεια παρατίθεται μία λειτουργική περιγραφή των ρουτινών που υλοποιούν τον οδηγό AAL.

### 3.3.3.1.1 Ρουτίνα Χειρισμού Διακοπών AAL

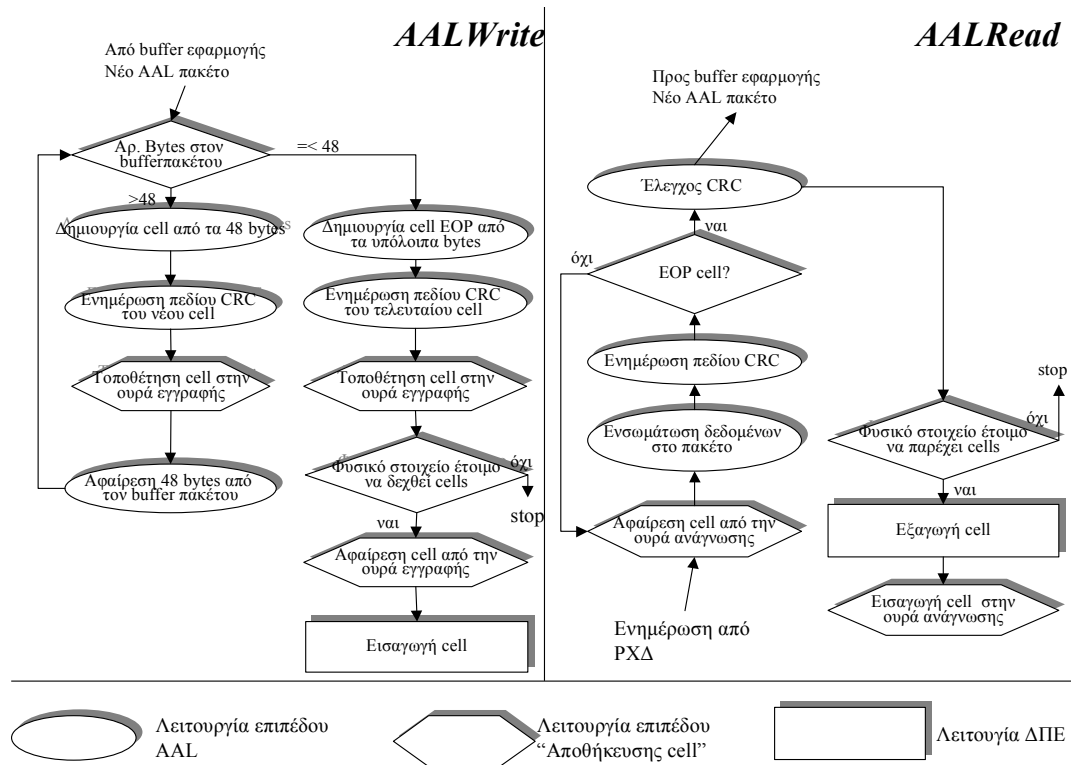
Το διάγραμμα ροής της P.X.Δ. φαίνεται στην Εικόνα 30.



**Εικόνα 30 Διάγραμμα ροής της P.X.Δ.**

Η P.X.Δ. καλείται όταν συμβεί μία διακοπή από το φυσικό στοιχείο της κάρτας. Όταν γίνει αντιληπτή η διακοπή η P.X.Δ. ελέγχει την αιτία της διαβάζοντας τον ειδικό καταχωρητή του φυσικού στοιχείου (ISR). Αν η διακοπή σημαίνει ότι το φυσικό στοιχείο είναι έτοιμο να δεχθεί ένα cell από τον επεξεργαστή τότε η P.X.Δ. εξάγει το cell που βρίσκεται στην κορυφή της ουράς εξόδου (βλ. Εικόνα 29) και μέσω κατάλληλης συνάρτησης της Δ.Χ.Φ.Σ. το εισάγει στο φυσικό στοιχείο και επιστρέφει. Αν η διακοπή σημαίνει ότι το φυσικό στοιχείο θέλει να εξάγει στον επεξεργαστή ένα cell διαχείρισης, τότε η P.X.Δ. μέσω κατάλληλης συνάρτησης της Δ.Χ.Φ.Σ. το εξάγει από το φυσικό στοιχείο και το τοποθετεί στην ουρά εισόδου (βλ. Εικόνα 29). Στη συνέχεια ελέγχει εάν το cell που εισήχθη ήταν το τελευταίο ενός AAL πακέτου. Αν ναι τότε πληροφορεί την εφαρμογή ότι υπάρχει ένα AAL πακέτο διαθέσιμο προς ανάγνωση. Αν όχι η P.X.Δ. επιστρέφει και αναμένει την επόμενη διακοπή.

### 3.3.3.1.2 Ρουτίνες AALWrite / AALRead



**Εικόνα 31 Διάγραμμα ροής Ρουτινών AALWrite, AALRead**

Η λειτουργία των ρουτινών AALRead και AALWrite του Οδηγού AAL φαίνεται στην Εικόνα 31.

**AALRead:** Η συγκεκριμένη ρουτίνα καλείται από την εφαρμογή υψηλότερου επιπέδου. Αρχικά εξάγει το πρώτο cell από την ουρά ανάγνωσης και ξεκινάει την δημιουργία του AAL πακέτου λαμβάνοντας τις απαραίτητες πληροφορίες από το payload του (σύμφωνα με το πρότυπο του AAL5 του ATM Forum). Στη συνέχεια ενημερώνει το CRC για το συγκεκριμένο cell και ελέγχει εάν το cell αυτό ήταν EOP. Αν όχι τότε εξάγει το επόμενο cell από την ουρά ανάγνωσης και επαναλαμβάνει την προηγούμενη διαδικασία. Όταν κάποια στιγμή προστεθεί στο πακέτο και το EOP cell τότε εκτελεί τον έλεγχο CRC για την ορθότητα του πακέτου και αποθηκεύει το πακέτο σε κατάλληλο buffer τον οποίο επιστρέφει. Πρωτού επιστρέψει ελέγχει εάν στο φυσικό στοιχείο της κάρτας υπάρχουν νεα cells προς εξαγωγή στον επεξεργαστή. Αν ναι τότε εκτελεί την διαδικασία εξαγωγής και τοποθέτησης στην ουρά (όμοια με την P.X.Δ.) και επιστρέφει. Αν όχι επιστρέφει αμέσως. Το τελευταίο βήμα πραγματοποιείται για να εξασφαλιστεί η περίπτωση, κατά την διάρκεια εκτέλεσης της ρουτίνας να έχει υπάρξει μία διακοπή από το φυσικό στοιχείο.

**AALWrite:** Η συγκεκριμένη ρουτίνα καλείται από την εφαρμογή υψηλότερου επιπέδου έχοντας ως είσοδο ένα AAL πακέτο. Αρχικά ελέγχει τον αριθμό των bytes του πακέτου. Αν αυτός είναι μικρότερος από 48 (πακέτο ενός cell) τότε δημιουργεί ένα cell διαχείρισης (VP=0, VC=5) που περιέχει τα bytes του πακέτου και το ορίζει ως EOP cell. Στη συνέχεια ενημερώνει το πεδίο CRC του cell και το τοποθετεί στην ουρά εγγραφής. Επίσης ελέγχει (στην περίπτωση που η διακοπή έχει έρθει ήδη αλλά

δεν ενεργοποιήθηκε η ΡΧΔ) αν το φυσικό στοιχείο είναι έτοιμο να δεχθεί cell οπότε σε περίπτωση θετικής απάντησης το εξάγει από την ουρά εγγραφής και το εισάγει στο φυσικό στοιχείο ενώ σε περίπτωση αρνητικής απάντησης επιστρέφει.

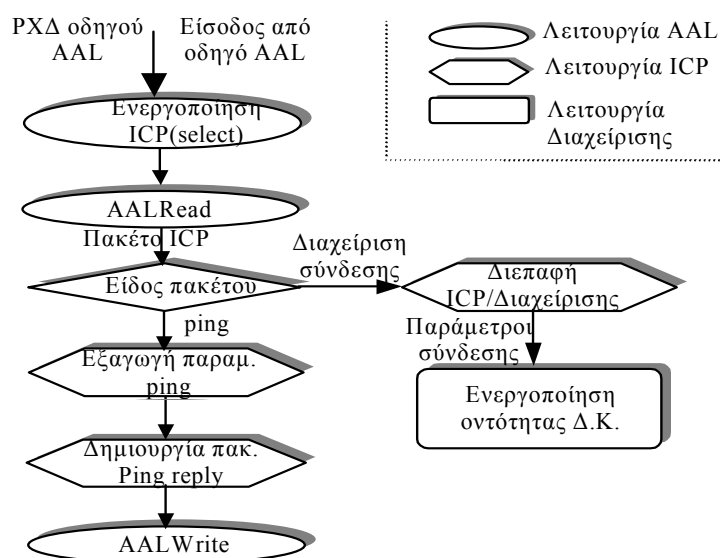
Αν ο αριθμός των bytes του πακέτου είναι μεγαλύτερος των 48, τότε η ρουτίνα δημιουργεί ένα cell διαχείρισης (VP=0, VC=5) από τα πρώτα 48 bytes, ενημερώνει το πεδίο CRC του cell και το τοποθετεί στην ουρά εγγραφής. Στη συνέχεια αφαιρεί τα 48 bytes από το πακέτο και ελέγχει ξανά τον αριθμό των εναπομεινόντων bytes. Αν ο αριθμός είναι πάλι μεγαλύτερος του 48 επαναλαμβάνει την διαδικασία. Αν είναι μικρότερος εκτελεί την δημιουργία EOP cell της προηγούμενης παραγράφου και αποστέλει αν είναι δυνατόν όλα τα cells του πακέτου στο φυσικό στοιχείο.

### 3.3.4 Εσωτερικό Πρωτόκολλο Επικοινωνίας (ICP)

Το Εσωτερικό Πρωτόκολλο Επικοινωνίας είναι η πρώτη εφαρμογή που εκτελείται έξω από τον πυρήνα του λειτουργικού συστήματος ως ανεξάρτητο πρόγραμμα (thread). Η εφαρμογή είναι σχεδιασμένη έτσι ώστε να υλοποιεί το κομμάτι του ICP που είναι απαραίτητο για την λειτουργία της κάρτας Εξυπηρετητή ABR. Πιο συγκεκριμένα υλοποιείται η επικοινωνία με την κάρτα Κεντρικού Επεξεργαστή μέσω των πακέτων ICP\_PING, ICP\_PING\_REPLY, ICP\_OPEN\_CONNECTION, ICP\_CLOSE\_CONNECTION, όπως περιγράφονται στην παράγραφο 1.6.

Η οντότητα ICP επικοινωνεί με τον οδηγό AAL και τον χρησιμοποιεί για την λήψη και αποστολή πακέτων διαχείρισης. Επίσης επικοινωνεί και χρησιμοποιεί το επίπεδο διαχείρισης και συγκεκριμένα την οντότητα διαχείρισης κλήσεων προκειμένου να εκτελεστούν οι εντολές εκκίνησης η κατάργησης νέων ροών που προέρχονται από τον Κεντρικό Επεξεργαστή.

Η λειτουργία της εφαρμογής ICP φαίνεται στην Εικόνα 32 και περιγράφεται στην επόμενη παράγραφο:



Εικόνα 32 Διάγραμμα Ροής λειτουργίας ICP

Η εφαρμογή ICP παραμένει ανενεργή και περιμένει να υπάρξει ενεργοποίηση του οδηγού AAL. Όταν υπάρξει είσοδος στον οδηγό (είσοδος πακέτου AAL) αυτός μέσω της P.X.Δ. «ξυπνάει» την εφαρμογή ICP (βλ. και Εικόνα 29). Μόλις το ICP αντιληφθεί την ύπαρξη ενός πακέτου εκτέλει εντολή write του οδηγού προκειμένου να διαβάσει το πακέτο. Αφού το διαβάσει εκτελείται το πρωτόκολλο και αναλύεται η επικεφαλίδα του ICP πακέτου. Αν το ICP πακέτο περιέχει μία εντολή PING τότε η εφαρμογή διαβάζει από τα περιεχόμενα του πακέτου PING τον αποστολέα του πακέτου. Στη συνέχεια συνθέτει ένα πακέτο ICP\_PING\_REPLY τοποθετώντας στην διεύθυνση προορισμού την διεύθυνση αποστολέα του ληφθέντος ICP\_PING πακέτου. Το συγκεκριμένο πακέτο το αποστέλει στον Κεντρικό Επεξεργαστή μέσω του οδηγού AAL εκτελώντας μία write εντολή.

Στη δεύτερη περίπτωση που το ICP πακέτο είναι πακέτο διαχείρισης συνδέσεων (ICP\_OPEN\_CONNECTION, ICP\_ABR\_CONNECTION), το ICP εκτελεί την λειτουργία διεπαφής με την οντότητα διαχείρισης συνδέσεων του Επιπέδου Διαχείρισης. Με αυτή τη λειτουργία το ICP αναλύει και εξάγει την πληροφορία σύνδεσης από το ICP πακέτο και την εισάγει σε μία δομή παραμέτρων σύνδεσης την οποία μπορεί να δεχθεί η οντότητα διαχείρισης συνδέσεων ως είσοδο. Τέλος η εφαρμογή ICP ενεργοποιεί το Επίπεδο Διαχείρισης, είτε μεταφέροντας την δομή παραμέτρων σε αυτό (εφόσον αυτό εκτελείται σε διαφορετικό thread) είτε καλώντας στο δικό της thread τις διαδικασίες διαχείρισης συνδέσεων που αποτελούν κομμάτι του Επιπέδου Διαχείρισης.

### 3.3.5 Οντότητες Διαχείρισης

Οι οντότητες διαχείρισης υλοποιούν το υψηλότερο επίπεδο του λογισμικού της κάρτας Εξυπηρετητή ABR. Εκτελούν τις λειτουργίες διαχείρισης όπως αυτές περιγράφονται στην παράγραφο 3.1.1.

Το Επίπεδο Διαχείρισης αποτελείται από τρεις οντότητες λογισμικού:

- *Οντότητα Αρχικοποίησης και Προγραμματισμού (Ο.Α.Π.):*
- *Οντότητα Διαχείρισης Συνδέσεων (Ο.Δ.Σ.)*
- *Οντότητα Ελέγχου και Διαχείρισης Σφαλμάτων (Ο.Ε.Δ.Σ.)*

Στις επόμενες παραγράφους περιγράφεται ο σχεδιασμός αυτών των οντοτήτων.

#### 3.3.5.1 Οντότητα Αρχικοποίησης και Προγραμματισμού (Ο.Α.Π.)

Αυτό το κομμάτι λογισμικού εκτελείται πρώτο κατά την εκκίνηση του συστήματος και για μία φορά μόνο. Εκτελεί τις διαδικασίες αρχικοποίησης του υλικού της κάρτας και προγραμματισμού του χρησιμοποιώντας τις Δ.Χ.Φ.Σ.. Αυτές οι διαδικασίες έχουν ως εξής:

- Εκκίνηση και αρχικοποίηση του Στοιχείου Μεταγωγής
- Προγραμματισμός του Στοιχείου Μεταγωγής ώστε να εκτελεί είτε την απλή λειτουργία διεπαφής στον κοινό δίαυλο κελιών, είτε να εκτελεί επιπλέον την



λειτουργία εξαγωγής/εισαγωγής κελιών από/προς τον επεξεργαστή στην περίπτωση που τα κελιά διαχείρισης μεταφέρονται μέσω αυτού του φυσικού στοιχείου. Επίσης ο προγραμματισμός του Στοιχείου Μεταγωγής καθορίζει το είδος της διεπαφής (UTOPIA8 ή 16) το οποίο το συνδέει με την Μ.Ε.Κ.Α.

- Εκκίνηση της Μονάδας Εξυπηρέτησης Κίνησης ABR και αρχικοποίηση των δομών αποθήκευσης κελιών σε ουρές στην μνήμη.
- Εκκίνηση και αρχικοποίηση του Επεξεργαστή Κελιών (εφόσον αυτό το φυσικό στοιχείο χρησιμοποιείται).
- Προγραμματισμός του Επεξεργαστή Κελιών έτσι ώστε:
  - Να επικοινωνεί με την Μ.Ε.Κ.Α. μέσω κατάλληλης διεπαφής UTOPIA (8 ή 16)
  - Να εισάγει και να εξάγει cells διαχείρισης ( $VP=0$ ,  $VC=31$ ) από και πρὸς τον επεξεργαστή (εφόσον η ανταλλαγή cells διαχείρισης γίνεται μέσω αυτού του φυσικού στοιχείου).
  - Να εισάγει και να εξάγει RM cells ( $PTI = 6$ ) από και πρὸς τον επεξεργαστή (εφόσον το λογισμικό υποστηρίζει λειτουργία Explicit Rate Marking).
  - Να πραγματοποιεί εισαγωγή κεφαλίδας δρομολόγησης για τον εσωτερικό δίαυλο κελιών (CellBus Routing Header) στα RM cells που δέχεται από τον επεξεργαστή και αποστέλει στην έξοδο του μεταγωγού (εφόσον το λογισμικό υποστηρίζει λειτουργία Explicit Rate Marking).

Μετά την αρχικοποίηση του υλικού η Ο.Α.Π. εκκινεί και αρχικοποιεί το λογισμικό της κάρτας θέτοντας το σε κατάσταση λειτουργίας. Πιο συγκεκριμένα:

- Συνδέει τις διακοπές που σχετίζονται με τα φυσικά στοιχεία με τις αντίστοιχες ρουτίνες εξυπηρέτησης διακοπών (ISRs)
- Εγκαθιστά τον οδηγό AAL και καθορίζει ποιά έκδοση του θα εκτελείται (πάνω από τον Επεξεργαστή Κελιών ή πάνω από το Στοιχείο Μεταγωγής).
- Εκκινεί τα threads των ρουτίνων χειρισμού διακοπών.
- Εκκινεί τα threads στα οποία θα εκτελούνται οι εφαρμογές υψηλότερου επιπέδου (ICP, Έλεγχος Ροής, υπόλοιπες οντότητες Διαχείρισης).

### 3.3.5.2 Οντότητα Διαχείρισης Συνδέσεων (Ο.Δ.Σ.)

Αυτή η οντότητα εκτελείται είτε αυτόνομα (ξεχωριστό thread) είτε καλούμενη από το πρωτόκολλο ICP το οποίο δέχεται τις εντολές διαχείρισης συνδέσεων από τον Κεντρικό Επεξεργαστή. Η επιλογή εξαρτάται από τον βαθμό του παραλληλισμού που θέλουμε να δώσουμε μεταξύ των λειτουργιών ICP και διαχείρισης συνδέσεων. Η Ο.Δ.Σ. δέχεται ως είσοδο μία δομή παραμέτρων σύνδεσης την οποία της παρέχει το

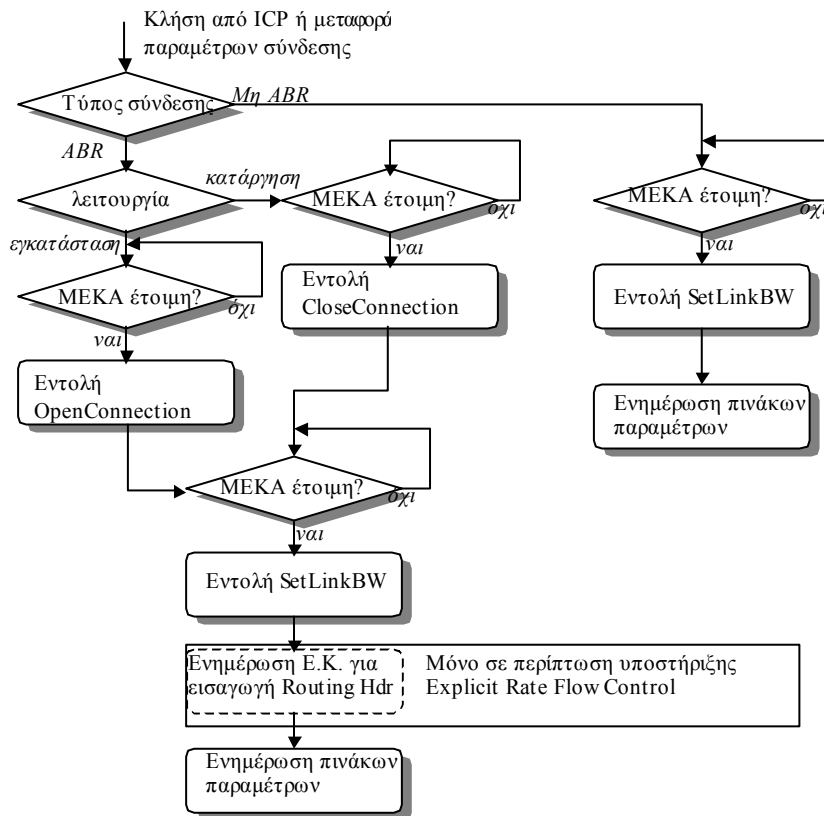
πρωτόκολλο ICP μέσω της διεπαφής τους (βλ. Παράγραφο 3.3.4). Οι παράμετροι αυτοί είναι:

- Τον τύπο της σύνδεσης (ABR – μη ABR)
- Την λειτουργία (εκκίνηση σύνδεσης / τερματισμός σύνδεσης)
- Τα VP/VC εισόδου/εξόδου της σύνδεσης
- Τα εσωτερικά TAGs των συνδέσεων (ICIs)
- Τα φυσικά links εισόδου/εξόδου της σύνδεσης
- Το διαθέσιμο για την ABR κυκλοφορία εύρος ζώνης (Available BW) σε κάθε φυσικό link μετά την εγκατάσταση ή κατάργηση της τρέχουσας σύνδεσης.
- Το Minimum Cell Rate της σύνδεσης στην περίπτωση που αυτή είναι τύπου ABR.

Στην συνέχεια με βάση αυτήν την δομή και χρησιμοποιώντας τις Δ.Χ.Φ.Σ. ενημερώνει τα φυσικά στοιχεία με τις απαραίτητες παραμέτρους για τον χειρισμό των cells που ανήκουν στις αιτούμενες συνδέσεις. Το φυσικό στοιχείο που απαιτεί το μεγαλύτερο μέρος της ενημέρωσης είναι η Μ.Ε.Κ.Α. Εκτός από αυτήν ενημέρωση χρειάζεται και ο Επεξεργαστής Κελιών αν και εφόσον η κάρτα υποστηρίζει Explicit Rate Flow Control.

Η Ο.Δ.Σ. ενημερώνει επίσης εσωτερικούς πίνακες στους οποίους περιέχονται παράμετροι που αφορούν τις συνδέσεις καθώς και τα links εισόδου/εξόδου του μεταγωγέα. Αυτές οι παράμετροι είναι αναγκαίοι καταρχήν για την εκτέλεση του αλγορίθμου ροής καθώς και για την διατήρηση τοπικής πληροφορίας που αφορά την κατάσταση του μεταγωγέα η οποία μπορεί να καταστεί χρήσιμη σε περίπτωση επέκτασης της λειτουργικότητας του.

Η αναλυτική λειτουργία της Ο.Δ.Σ. φαίνεται στην Εικόνα 33 και περιγράφεται στην επόμενη παράγραφο.



Εικόνα 33 Διάγραμμα Ροής Ο.Δ.Σ.

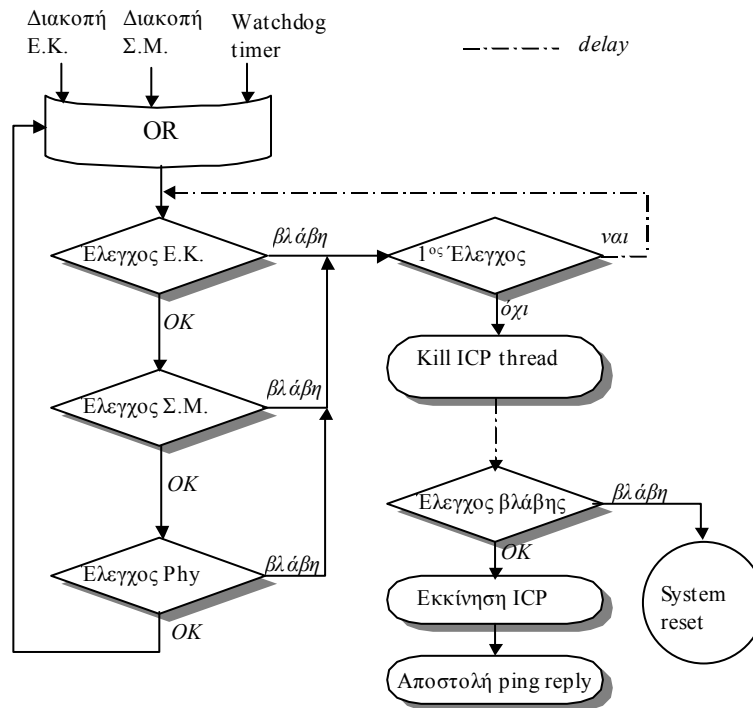
Η Ο.Δ.Σ. καλείται από το thread του ICP με είσοδο την δομή παραμέτρων σύνδεσης, ή εναλλακτικά ως ανεξάρτητο thread ενεργοποιείται από το thread του ICP και το τελευταίο του μεταβιβάζει την δομή παραμέτρων σύνδεσης μέσω μίας λειτουργίας intertask communication. Αρχικά η Ο.Δ.Σ. ελέγχει τον τύπο της σύνδεσης που περιέχεται στην δομή παραμέτρων. Εάν η σύνδεση είναι τύπου ABR τότε ο Ο.Δ.Σ. ελέγχει την λειτουργία της αίτησης (εγκατάσταση ή κατάργηση σύνδεσης). Αν η αίτηση είναι για εγκατάσταση σύνδεσης τότε αρχίζει η ενημέρωση των φυσικών στοιχείων. Αρχικά ενημερώνεται η Μ.Ε.Κ.Α. Η Ο.Δ.Σ. ελέγχει εάν η Μ.Ε.Κ.Α. είναι έτοιμη να δεχθεί μία εντολή από τον επεξεργαστή. Αν δεν είναι η Ο.Δ.Σ. πραγματοποιεί polling έως ότου η Μ.Ε.Κ.Α. καταστεί έτοιμη. Όταν συμβεί αυτό η Ο.Δ.Σ. παρέχει στην Μ.Ε.Κ.Α. εντολή OpenConnection. Αμέσως μετά και όταν η Μ.Ε.Κ.Α. καταστεί έτοιμη για εντολή ξανά η Ο.Δ.Σ. της παρέχει εντολή SetLinkBW για να την ενημερώσει για το διαθέσιμο εύρος ζώνης μετά την εγκατάσταση της σύνδεσης. Το επόμενο βήμα είναι η ενημέρωση του Ε.Κ. έτσι ώστε αυτός να τοποθετεί το κατάλληλο Routing Header στα RM cells που ανήκουν στην προς εγκατάσταση σύνδεση και προέρχονται από τον επεξεργαστή (εισερχόμενα RM cells). Τέλος η Ο.Δ.Σ. ενημερώνει και τους εσωτερικούς πίνακες παραμέτρων που διατηρεί με τις παραμέτρους της νέας σύνδεσης.

Εαν η αίτηση είναι για κατάργηση μίας ABR σύνδεσης τότε ακολουθείται ολοκληρη η προηγούμενη διαδικασία με την διαφορά ότι τώρα η Ο.Δ.Σ. παρέχει πρώτα μία εντολή CloseConnection στην Μ.Ε.Κ.Α.

Τέλος αν η νέα σύνδεση είναι μη ABR τύπου, τότε ανεξάρτητα αν πραγματοποιείται εγκατάσταση ή κατάργηση η δομή παραμέτρων περιέχει μόνο τα ανανεωμένα διαθέσιμα εύρη ζώνης των φυσικών links στα οποία ανήκει η σύνδεση. Έτσι η Ο.Δ.Σ. το μόνο που έχει να κάνει είναι να παρέχει στην Μ.Ε.Κ.Α. μία εντολή SetLinkBW με τα νέα εύρη ζώνης και να ενημερώσει τους εσωτερικούς πίνακες σχετικά με αυτά.

**3.3.5.3 Οντότητα Ελέγχου και Διαχείρισης Σφαλμάτων (Ο.Ε.Δ.Σ.)**

Η Ο.Ε.Δ.Σ. έχει την ευθύνη της ορθής λειτουργίας των φυσικών στοιχείων της κάρτας. Διενεργεί ελέγχους κατά τακτά χρονικά διαστήματα χρησιμοποιώντας rolling ή ενεργοποιείται από διακοπές που παράγουν τα φυσικά στοιχεία της κάρτας και διαβάζοντας τους ειδικευμένους καταχωρητές κατάστασης (Status Registers) εξακριβώνει την ορθή λειτουργία των φυσικών στοιχείων. Σε περίπτωση που υποπέσει στην αντίληψη της Ο.Ε.Δ.Σ. ένα σφάλμα σε κάποιο φυσικό στοιχείο αυτή αναλαμβάνει να εκτελέσει κατάλληλες διαδικασίες για τον χειρισμό της κατάστασης. Η αναλυτική λειτουργία της Ο.Ε.Δ.Σ. φαίνεται στην Εικόνα 34 και περιγράφεται στην επόμενη παράγραφο.



**Εικόνα 34 Διάγραμμα Ροής Ο.Ε.Δ.Σ.**

Η Ο.Ε.Δ.Σ. ενεργοποιείται από διακοπές οι οποίες είναι είτε εξωτερικές (προερχόμενες από τα φυσικά στοιχεία) είτε εσωτερικές προερχόμενες από έναν watchdog timer. Στην πρώτη περίπτωση ο έλεγχος πραγματοποιείται ασύγχρονα όταν συμβεί ένα σφάλμα ενώ στην δεύτερη πραγματοποιείται περιοδικός έλεγχος στα φυσικά στοιχεία ανά ορισμένο χρονικό διάστημα.

Όταν συμβεί μία από τις δύο πιο πάνω περιπτώσεις (μία διακοπή, ή περιοδικό έλεγχο) ενεργοποιείται ο Ο.Ε.Δ.Σ. η οποία πραγματοποιεί έλεγχο σε όλα τα φυσικά

στοιχεία με την σειρά (Επεξεργαστής Κελιών, Στοιχείο Μεταγωγής, Στοιχείο ΡΗΥ). Σε περίπτωση που δεν ανιχνευθεί κάποιο σφάλμα (η ανίχνευση γίνεται μέσω του Status Register κάθε στοιχείου), η σειρά ελέγχου τερματίζεται μέχρι το επόμενο event ελέγχου.

Εάν ανιχνευθεί κάποιο σφάλμα σε ένα φυσικό στοιχείο τότε οι δράσεις που αναλαμβάνει η Ο.Ε.Δ.Σ. είναι οι εξής:

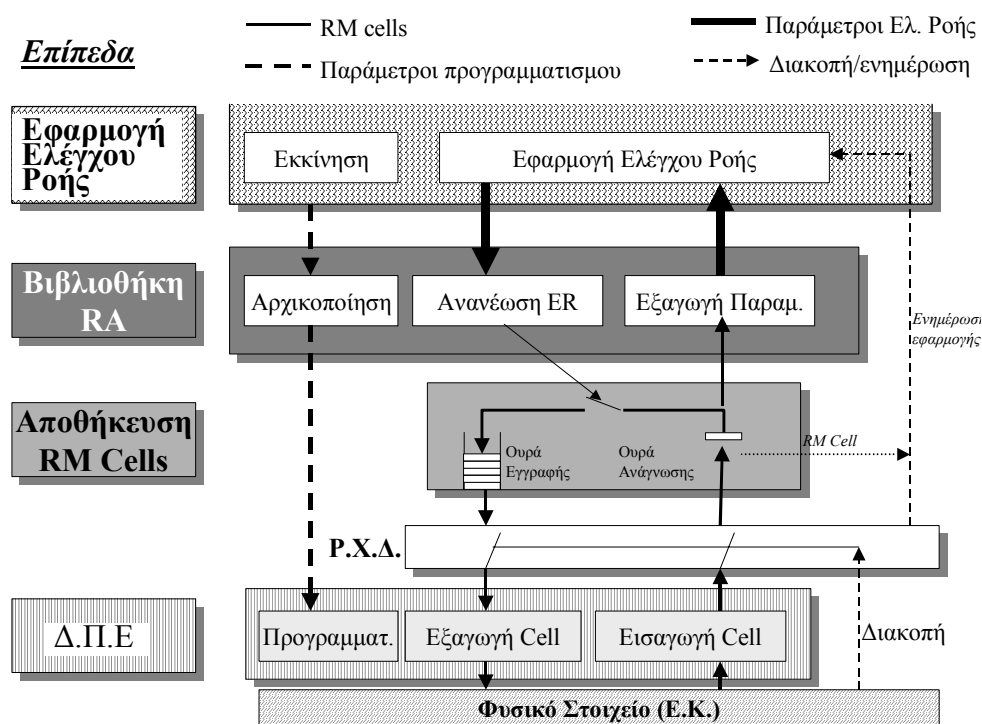
- Αρχικά καθυστερεί κατά κάποιο συγκεκριμένο χρονικό διάστημα και ελέγχει για δεύτερη φορά το προβληματικό φυσικό στοιχείο για να δει εάν το σφάλμα ήταν παροδικό.
- Αν και ο δεύτερος έλεγχος δείξει το ίδιο σφάλμα τότε η Ο.Ε.Δ.Σ. θεωρεί ότι η κάρτα εκτελεί εσφαλμένη λειτουργία. Για αυτόν τον λόγο εκτελεί το πρώτο βήμα διαχείρισης το οποίο είναι ο τερματισμός του thread του ICP. Αυτό έχει ως σκοπό να διακόψει την αποστολή πακέτων PING REPLY προς τον Κεντρικό Επεξεργαστή έτσι ώστε ο τελευταίος να θεωρήσει ότι η κάρτα βρίσκεται προσωρινά εκτός λειτουργίας και να αναλάβει τις δράσεις που πρέπει (π.χ. να διακόψει προσωρινά τις ABR συνδέσεις).
- Αμέσως μετά η Ο.Ε.Δ.Σ. καθυστερεί για ένα σχετικά μεγάλο χρονικό διάστημα (της τάξης των seconds) και αμέσως μετά πραγματοποιεί έναν τρίτο έλεγχο στο προβληματικό στοιχείο.
- Αν ο τρίτος έλεγχος αποβεί «αρνητικός» (το σφάλμα έπαψε να υπάρχει και το φυσικό στοιχείο επανήλθε) τότε η Ο.Ε.Δ.Σ. θεωρεί ότι η δυσλειτουργία της κάρτας δεν είναι μόνιμη και μη αντιστρέψιμη οπότε επανεκκινεί το thread του ICP και αμέσως στέλνει ένα πακέτο PING REPLY στον Κεντρικό Επεξεργαστή για να τον ενημερώσει για την επαναφορά της κάρτας. Ο Κεντρικός Επεξεργαστής μπορεί συνεπώς να ρυθμιστεί ώστε να θεωρεί την κάρτα μόνιμα νεκρή μετά από χρόνο ο οποίος είναι μεγαλύτερος από τον χρόνο του τρίτου ελέγχου της τοπικής Ο.Ε.Δ.Σ. και εφόσον λάβει PING REPLY σε αυτόν τον χρόνο να πράξει αναλόγως
- Αν τέλος ο τρίτος έλεγχος αποβεί και αυτός θετικός τότε η Ο.Ε.Δ.Σ. προκαλεί ένα reset της κάρτας η οποία πλέον λειτουργεί από την αρχή. Αυτό σημαίνει ότι οι συνδέσεις που υπήρχαν κατά την εμφάνιση του σφάλματος χάνονται μόνιμα και αναμένονται νέες.

Οι παραπάνω δράσεις πραγματοποιούνται στις περιπτώσεις αναστρέψιμων σφαλμάτων οι οποίες δεν επηρεάζουν (αν υφίστανται για σχετικά μικρό χρονικό διάστημα) την ορθή λειτουργία του συστήματος (π.χ. λανθασμένη λήψη ενός cell από το Στοιχείο Μεταγωγής δεν σημαίνει ότι το cell χάθηκε, αλλά θα ξαναμεταδοθεί πάνω από τον κοινό δίαυλο κελιών). Στην περίπτωση μη αναστρέψιμων σφαλμάτων (π.χ. απώλεια cells διαχείρισης από τον Επεξεργαστή Κελιών) τότε οι διαδικασίες είναι συνοπτικές: δεν πραγματοποιείται δεύτερος και τρίτος έλεγχος αλλά αμέσως η κάρτα απενεργοποιείται μέχρις ότου ο Κεντρικός Επεξεργαστής την θεωρήσει μόνιμα νεκρή, και στην συνέχεια πραγματοποιείται επανεκκίνηση της κάρτας και εκ νέου ενσωμάτωσή της στην λειτουργικότητα του μεταγωγέα.

### 3.3.6 Βιβλιοθήκη RA (RA Library)

Η βιβλιοθήκη RA σχεδιάστηκε για να υλοποιεί το επίπεδο RA (RM cell Adaption Layer). Αυτή η βιβλιοθήκη περιέχει συναρτήσεις οι οποίες χειρίζονται τα RM cells που εισάγονται στον επεξεργαστή κυρίως αναλύοντας τα πεδία τους και εξάγοντας την απαραίτητη πληροφορία που απαιτείται από τον αλγόριθμο Ελέγχου Ροής του Υψηλότερου Επιπέδου. Καθώς ο Έλεγχος Ροής είναι μία βασική λειτουργία που πρέπει να εκτελείται από μία κάρτα Εξυπηρετητή ABR, αυτή η βιβλιοθήκη είναι απαραίτητο να υπάρχει ως μόνιμο κομμάτι του λειτουργικού της κάρτας. Επίσης η υλοποίηση αυτών των λειτουργιών με μορφή βιβλιοθήκης παρέχει την απαραίτητη ανεξαρτησία των διαδικασιών χειρισμού των RM cells, οι οποίες χαρακτηρίζονται από γενικότητα, από τους αλγορίθμους Ελέγχου Ροής που τις χρησιμοποιούν και οι οποίοι μπορούν να μεταβάλλονται σε διαφορετικούς χώρους και συνθήκες λειτουργίας του συστήματος.

Εξαιτίας του γεγονότος ότι τα RM cells εισάγονται στον επεξεργαστή μόνο μέσω της διεπαφής που παρέχεται από τον Επεξεργαστή Κελιών, η συγκεκριμένη βιβλιοθήκη χρησιμοποιεί την ΔΧΦΣ αυτού του φυσικού στοιχείου. Οι λειτουργίες που παρέχει στην εφαρμογή Ελέγχου Ροής η συγκεκριμένη βιβλιοθήκη αναλυτικά φαίνονται στην Εικόνα 35 και περιγράφονται ως εξής:



Εικόνα 35 Λειτουργία και μεταφορά πληροφορίας Ελέγχου Ροής

- *Ρουτίνα Χειρισμού Διακοπών(P.X.Δ.):* Όπως και στον οδηγό AAL έτσι και στην περίπτωση του Ελέγχου Ροής, όλες οι λειτουργίες πραγματοποιούνται σε cells (RM cells) τα οποία εισάγονται στον επεξεργαστή μέσω των φυσικών στοιχείων (συγκεκριμένα μέσω του Επεξεργαστή Κελιών). Για αυτόν τον λόγο θα πρέπει να υποστηρίζεται από το Επίπεδο RA μία ρουτίνα χειρισμού διακοπών που

προέρχονται από το φυσικό στοιχείο όταν αυτό είναι έτοιμο να εισάγει ή να εξάγει ένα cell από ή προς τον επεξεργαστή. Η εισαγωγή και η εξαγωγή γίνεται μέσω του Δ.Π.Ε.

- *Αποθήκευση Cells*: Η βιβλιοθήκη RA πρέπει να είναι προσαρμοσμένη στο επίπεδο αποθήκευσης cells όπως και ο οδηγός AAL. Κατά την αρχικοποίηση δημιουργούνται δύο ουρές (εισόδου/εξόδου) για τα RM cells.
- *Αρχικοποίηση*: Η αρχικοποίηση περιέχει ρουτίνες που χρησιμοποιούνται για τον προγραμματισμό του φυσικού στοιχείου έτσι ώστε αυτό να είναι σε θέση να εισάγει και να εξάγει RM cells.
- *Εξαγωγή Παραμέτρων*: Η ρουτίνες εξαγωγής παραμέτρων χρησιμοποιείται από την εφαρμογή ελέγχου ροής προκειμένου να εξαχθούν οι παράμετροι ελέγχου ροής από τα εισερχόμενα RM cells. Αυτές οι παράμετροι τοποθετούνται σε κατάλληλες δομές οι οποίες αποτελούν είσοδο στην εφαρμογή Ελέγχου Ροής και με την χρήση αυτών εκτελείται ο επιλεγμένος αλγόριθμος. Με αυτές τις ρουτίνες επίσης υλοποιείται και η λειτουργία μετατροπής της ποσοτικής αναπαράστασης των διαφόρων παραμέτρων (κυρίως τιμές ρυθμών μετάδοσης) από την αναπαράσταση που χρησιμοποιείται στα πεδία των RM cells (floating point) σε αναπαράσταση κατάλληλη για την εφαρμογή Ελέγχου Ροής (integer, ή floating point του MPC860).
- *Ανανέωση ER*: Τέλος υπάρχουν η ρουτίνες οι οποίες εγγράφουν στα RM cells τις ανανεωμένες (από τον αλγόριθμο) τιμές του πεδίου ER και φροντίζουν να προωθήσουν το ανανεωμένο RM cell προς την έξοδο.

Η λειτουργία των επιπέδων που βρίσκονται κάτω από την εφαρμογή Ελέγχου Ροής έχει ως εξής (βλ. Εικόνα 35):

Όταν φτάνει ένα RM cell στον Επεξεργαστή Κελιών αυτός ενημερώνει τον επεξεργαστή με μία διακοπή. Αυτή η διακοπή ενεργοποιεί την P.X.Δ. η οποία δεν κάνει τίποτα άλλο από το να διαβάσει το cell από τον E.K. να το εισάγει στην ουρά εισόδου και να ειδοποιεί την εφαρμογή Ελέγχου Ροής ότι ένα νέο RM cell είναι διαθέσιμο προς επεξεργασία. Στην συνέχεια η Εφαρμογή χρησιμοποιώντας ρουτίνες του επιπέδου RA εξάγει τις απαραίτητες παραμέτρους από το RM cell (Connection TAG,CCR, ER) και εισάγοντάς τις σε κατάλληλες δομές τις χρησιμοποιεί για την εκτέλεση του αλγορίθμου.

Στο τέλος ο αλγόριθμος υπολογίζει την ανανεωμένη τιμή ER και χρησιμοποιώντας πάλι ρουτίνα της βιβλιοθήκης RA (Ανανέωση ER) το εισάγει στο RM cell το οποίο επεξεργάζεται (αυτό βρίσκεται ακόμη στην κορυφή της ουράς εισόδου). Στην συνέχεια η ίδια ρουτίνα RA αναλαμβάνει να εξάγει το ανανεωμένο RM cell από την ουρά εισόδου και να το προωθήσει στην ουρά εξόδου. Από εκεί μεταφέρεται στον E.K. μέσω της Ρουτίνας Χειρισμού Διακοπών. Τέλος ο E.K. εισάγει στο RM cell την επικεφαλίδα δρομολόγησης πάνω από τον κοινό δίαυλο κελιών (Routing Header) και το προωθεί προς την κάρτα εξόδου.

Ο προγραμματισμός του E.K. για την εισαγωγή της επικεφαλίδας δρομολόγησης πραγματοποιείται κατά την εκκίνηση κάθε νέας ABR σύνδεσης από το επίπεδο Διαχείρισης.

### 3.3.7 Εφαρμογή Ελέγχου Ροής

Η Εφαρμογή Ελέγχου Ροής αποτελεί την υλοποίηση του αλγορίθμου Ελέγχου Ροής Explicit Rate Marking που επιλέχθηκε να τρέχει στην κάρτα Εξυπηρετητή ABR. Η ρουτίνα που υλοποιεί αυτόν τον αλγόριθμο δέχεται ως είσοδο μία δομή παραμέτρων ελέγχου ροής που προέρχονται από τα εκάστοτε RM cells προς επεξεργασία και εξάγονται μέσω ρουτινών της βιβλιοθήκης RA. Αυτές οι παράμετροι που είναι απαραίτητες για την εκτέλεση του αλγορίθμου που επιλέχθηκε είναι:

- Το εσωτερικό αναγνωριστικό (TAG) που περιγράφει σε ποιά σύνδεση ανήκει ένα RM cell (από εδώ και στο εξής θα ονομάζεται FlowId(FId))
- Η τιμή Current Cell Rate (CCR) που περιέχεται στο αντίστοιχο πεδίο των RM cells (βλ. Παράρτημα Β)
- Η τιμή Explicit Rate (ER) που περιέχεται στο αντίστοιχο πεδίο των RM cells(βλ. Παράρτημα Β)

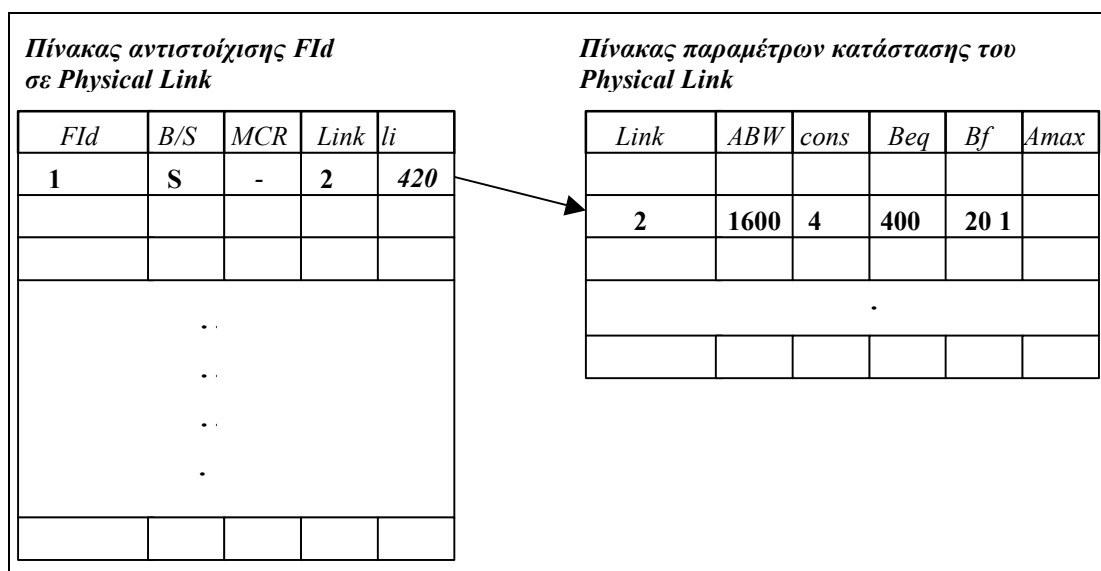
Εκτός από τις παραπάνω παραμέτρους που βρίσκονται στο εσωτερικό των RM cells, απαιτούνται για την εκτέλεση του αλγορίθμου και παράμετροι που περιγράφουν την κατάσταση του link εξόδου (βλ. Παράρτημα Α.2). Αυτοί οι παράμετροι περιέχονται σε εσωτερικούς πίνακες που διατηρεί το λογισμικό και ενημερώνονται από την Οντότητα Διαχείρισης Συνδέσεων του Επιπέδου Διαχείρισης κάθε φορά που εγκαθίσταται ή καταργείται μία σύνδεση. Οι πίνακες που είναι απαραίτητοι για την εκτέλεση του αλγορίθμου είναι δύο:

- *Ο πίνακας αντιστοίχισης FId σε Physical Link:* Μέσω αυτού του πίνακα γίνεται η αντιστοίχιση του FId ενός cell με το Physical link στο οποίο ανήκει αυτή η σύνδεση. Επίσης σε αυτόν τον πίνακα περιέχονται και κάποια στοιχεία για το κάθε FId όπως το Minimum Cell Rate του, η κατάστασή του (Bottlenecked/Satisfied, βλ. Παράρτημα Α.2) καθώς και ο τρέχων ρυθμός μετάδοσης li (το CCR του τελευταίου cell που υπέστη επεξεργασία).
- *Ο πίνακας παραμέτρων κατάστασης του Physical Link:* Αυτός ο πίνακας περιέχει για κάθε Physical link από το οποίο διέρχεται μία σύνδεση ABR, παραμέτρους που περιγράφουν την κατάσταση του και οι οποίες είναι αναγκαίες για την εκτέλεση του αλγορίθμου Ελέγχου Ροής. Αυτοί οι παράμετροι είναι:
  - *ABW:* Το διαθέσιμο εύρος ζώνης που παρέχεται από το Physical Link για την κίνηση ABR. Αυτή η παράμετρος περιέχεται στις παραμέτρους σύνδεσης του επιπέδου διαχείρισης και ενημερώνεται κατά την εγκατάσταση ή κατάργηση μίας νέας σύνδεσης οποιουδήποτε τύπου.



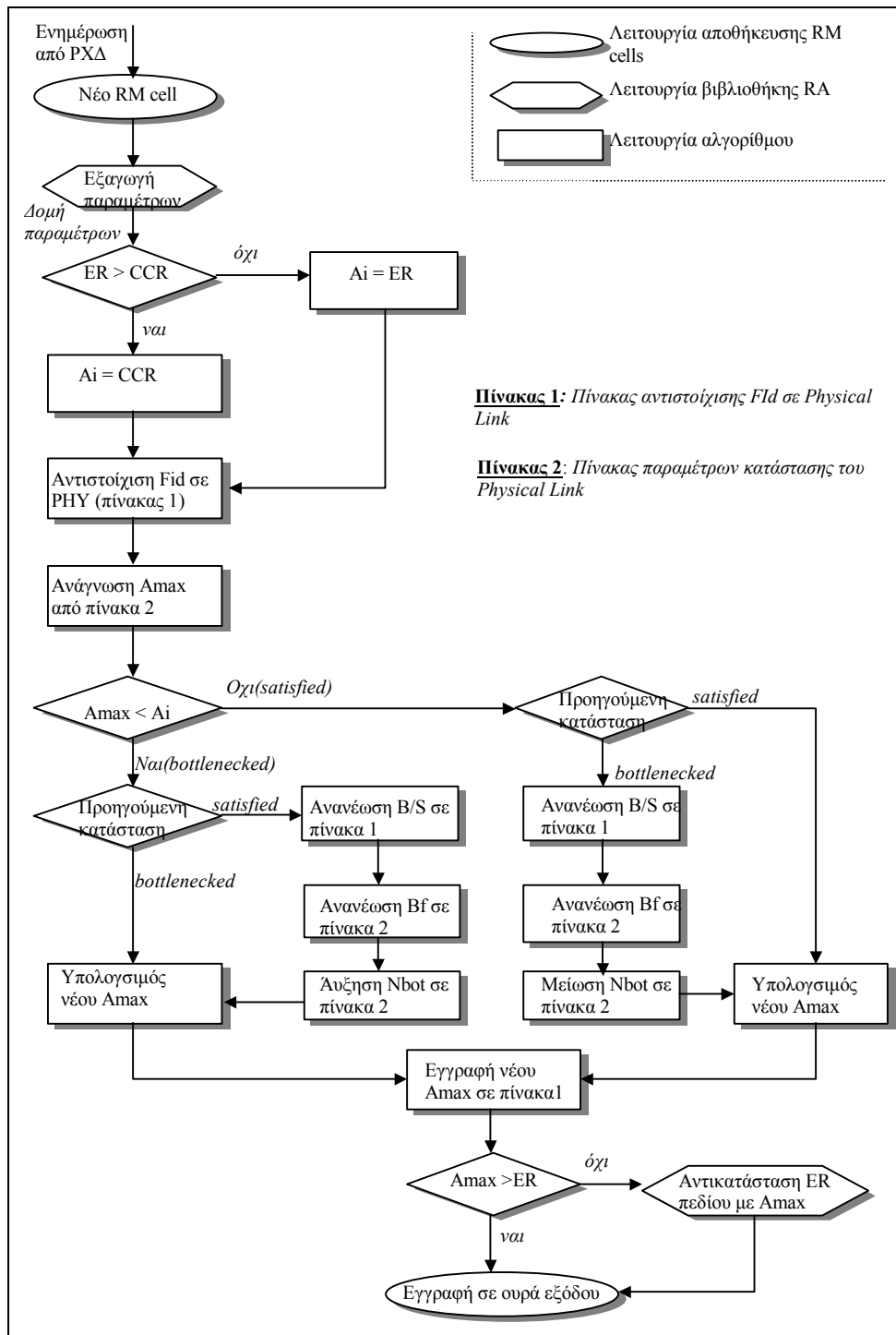
- *Cons*: Ο αριθμός των ABR συνδέσεων που διέρχονται από το Physical Link. Αυτός ο αριθμός ανανεώνεται κάθε φορά που εγκαθίσταται ή καταργείται μία ABR σύνδεση.
- *Beq*: Το fairshare του διαθέσιμου εύρους ζώνης. Προκύπτει από την διαίρεση του ABW προς το cons.
- *Bf*: (βλ Παράρτημα A.2). Η τιμή αυτής της παραμέτρου ανανεώνεται μετά την εκτέλεση του αλγορίθμου
- *Nbot* (βλ Παράρτημα A.2). Η τιμή αυτής της παραμέτρου ανανεώνεται μετά την εκτέλεση του αλγορίθμου
- *Amax* (βλ Παράρτημα A.2). Η τιμή αυτής της παραμέτρου ανανεώνεται μετά την εκτέλεση του αλγορίθμου και αποτελεί την έξοδο της εφαρμογής

Η δομή των παραπάνω πινάκων και η συσχέτιση μεταξύ τους φαίνεται στην Εικόνα 36



**Εικόνα 36** Εσωτερικοί πίνακες παραμέτρων για τον έλεγχο ροής

Η υλοποίηση του αλγορίθμου με βάση τις παραμέτρους των RM cells και τους πίνακες που περιγράφηκαν παραπάνω φαίνεται στην Εικόνα 37 και αναλύεται στην επόμενη παράγραφο.



Εικόνα 37 Διάγραμμα Ροής Υλοποίησης Αλγορίθμου Ροής

### Περιγραφή Υλοποίησης Αλγορίθμου Ελέγχου Ροής (A.E.P.)

Η υλοποίηση του αλγορίθμου βασίζεται στην περιγραφή που δόθηκε στο Παράρτημα Β.

Η ρουτίνα του A.E.P. καλείται ή εκκινείται από την Ρ.Χ.Δ. όταν η τελευταία εισάγει στον επεξεργαστή ένα RM cell. Η A.E.P. αμέσως διαβάζει το cell και χρησιμοποιώντας ρουτίνα RA εξάγει σε μία δομή τις απαραίτητες παραμέτρους από

το cell. Αυτές είναι το FId, το πεδίο ER και το πεδίο CCR. Με αυτή τη δομή και συμβουλευόμενος τους πίνακες 1 και 2 (1:πίνακας αντιστοίχισης FId σε Physical Link, 2: πίνακας παραμέτρων κατάστασης Physical link) ξεκινάει η εκτέλεση του αλγορίθμου.

Πρώτα πραγματοποιείται μετατροπή της τιμής των ER και CCR από την αναπαράσταση που τοποθετείται στα RM cells στην αναπαράσταση που χρησιμοποιεί ο αλγόριθμος και μετά συγκρίνονται οι τιμές ER και CCR. Η τιμή που είναι μικρότερη θεωρείται από τον αλγόριθμο ως ο τρέχων ρυθμός μετάδοσης της συγκεκριμένης σύνδεσης,  $A_i$  (βλ. Παράρτημα B.2).

Στη συνέχεια από τον πίνακα 1 και το FId γίνεται η αντιστοίχιση της σύνδεσης στην οποία ανήκει το cell με το Physical Link εξόδου της. Με το Physical Link σαν δείκτη διαβάζεται το τρέχον  $A_{max}$  (βλ. Παράρτημα B.2) από τον πίνακα 2. Μετά γίνεται σύγκριση μεταξύ του  $A_{max}$  και του  $A_i$ . Αν το  $A_{max}$  είναι μικρότερο τότε η σύνδεση θεωρείται bottlenecked από τον αλγόριθμο, αλλιώς θεωρείται satisfied (βλ. Παράρτημα B.2).

Και στις δύο περιπτώσεις τα επόμενα βήματα είναι παρόμοια: Πρώτα ελέγχεται η προηγούμενη κατάσταση της σύνδεσης από την εγγραφή B/S (Bottlenecked/Satisfied) πίνακα 1. Αν είναι ίδια με την τωρινή τότε κατευθείαν υπολογίζεται το νέο  $A_{max}$  από την κατάλληλη εξίσωση (βλ. Παράρτημα B.2). Οι παράμετροι για τον υπολογισμό της εξίσωσης ( $B_{eq}$ ,  $B_f$ ,  $N_{bot}$ ) λαμβάνονται απευθείας από τον πίνακα 2 ως τα περιεχόμενα στην θέση που δείχνει το Physical Link.

Αν η προηγούμενη κατάσταση είναι διαφορετική από την τωρινή τότε:

- Αναεώνεται το πεδίο B/S του πίνακα 1 στην τρέχουσα κατάσταση
- Αναεώνεται η τιμή  $B_f$  στον πίνακα 2 για το συγκεκριμένο Physical Link
- Αναεώνεται η τιμή  $N_{bot}$  στον πίνακα 2 για το συγκεκριμένο Physical Link. Αν έχουμε αλλαγή από satisfied σε bottlenecked το  $N_{bot}$  αυξάνεται κατά 1. Στην άλλη περίπτωση μειώνεται κατά 1.
- Υπολογίζεται το νέο  $A_{max}$  από την κατάλληλη εξίσωση (βλ. Παράρτημα B.2). Ξανά οι παράμετροι για τον υπολογισμό της εξίσωσης ( $B_{eq}$ ,  $B_f$ ,  $N_{bot}$ ) λαμβάνονται απευθείας από τον πίνακα 2 ως τα περιεχόμενα στην θέση που δείχνει το Physical Link.

Στο τέλος της εκτέλεσης του αλγορίθμου το νέο  $A_{max}$  εγγράφεται στον στην αντίστοιχη θέση του πίνακα 2 για το συγκεκριμένο Physical Link ώστε να χρησιμοποιηθεί την επόμενη φορά που θα έρθει RM cell που ανήκει σε σύνδεση αυτού του link. Αμέσως μετά πραγματοποιείται μετατροπή της τιμής του  $A_{max}$  από την αναπαράσταση που χρησιμοποιεί ο αλγόριθμος στην αναπαράσταση που τοποθετείται στα RM cells και σύγκριση της τιμής ER στο αντίστοιχο πεδίο του RM cell και της τιμής  $A_{max}$  που έχει προκύψει από τον αλγόριθμο. Αν το  $A_{max}$  είναι μικρότερο της τρέχουσας τιμής ER τότε το πεδίο ER του RM cell αντικαθίσταται από

την τιμή Amax. Αν το Amax που έχει προκύψει είναι μεγαλύτερο τότε δεν πραγματοποιείται αντικατάσταση του ER.

Τέλος το RM cell τοποθετείται στην ουρά εξόδου και η P.X.Δ. είναι υπεύθυνη να το προωθήσει στον Επεξεργαστή Κελιών όταν αυτός είναι έτοιμος να δεχθεί cells, και από εκεί στην κάρτα και το link εξόδου, αφού πρώτα τοποθετηθεί σε αυτό μία κεφαλίδα δρομολόγησης στον Κοινό Δίαυλο Κελιών (Routing Header) από την μονάδα μετάφρασης του E.K.

### 3.3.8 Οργάνωση και Ολοκλήρωση των οντοτήτων λογισμικού

#### 3.3.8.1 Γενική Οργάνωση

Το τελευταίο βήμα στον σχεδιασμό και την υλοποίηση του λογισμικού της κάρτας Εξυπηρετητή ABR είναι η ολοκλήρωση των διαφόρων οντοτήτων μεταξύ τους και με το λειτουργικό σύστημα καθώς και η οργάνωση της εκτέλεσης τους σε ένα ενιαίο σύστημα. Ο multitasking χαρακτήρας του λειτουργικού συστήματος μας επιτρέπει να οργανώσουμε τις οντότητες του λογισμικού σε ξεχωριστά tasks τα οποία επιτρέπουν τον παραλληλισμό της εκτέλεσης των διαφόρων διεργασιών και μέσω των δυνατοτήτων χρονοπρογραμματισμού που παρέχονται είναι δυνατόν να υλοποιηθούν αποδοτικά λειτουργίες πραγματικού χρόνου, όπου αυτό είναι απαραίτητο.

Με βάση της απαιτήσεις και την λειτουργικότητα του συστήματος, οι διεργασίες του λογισμικού της κάρτας οργανώθηκαν με τον εξής τρόπο:

- Χρησιμοποιήθηκαν ρουτίνες εξυπηρέτησης διακοπών (ISRs) η οποίες συνδέθηκαν (μέσω κλήσεων συστήματος) με τις φυσικές διακοπές του φυσικών στοιχείων. Στο VxWorks αυτές οι ρουτίνες εκτελούνται πάντα αυτόνομα (έξω από οποιοδήποτε context) καθιστώντας έτσι την απόκριση του συστήματος σε διακοπές ταχύτατη.
- Οι Ρουτίνες Χειρισμού Διακοπων (P.X.Δ.) των διαφόρων φυσικών στοιχείων και των οδηγών του συστήματος, οργανώνονται ως ξεχωριστά tasks η κάθε μία. Αυτό πραγματοποιήθηκε γιατί οι P.X.Δ. σε μερικές περιπτώσεις σχετίζονται με περισσότερες της μίας εφαρμογής υψηλότερου επιπέδου. Συνεπώς αν αυτές οι εφαρμογές τρέχουν σε διαφορετικά tasks δεν είναι δυνατόν να εξυπηρετηθούν από μία P.X.Δ. εκτός και αν αυτή εκτελείται ταυτόχρονα
- Οι οντότητες υψηλού επιπέδου (Διαχείριση, Έλεγχος Ροής) οργανώθηκαν ως εξής:
  1. Η διαχείριση συνδέσεων υλοποιήθηκε ως ένα ενιαίο task μέσα στο οποίο εκτελούνται ο Οδηγός AAL, το πρωτόκολλο ICP και η Ο.Δ.Σ. Ο λόγος για την αυτήν την ενιαία οργάνωση είναι ο υψηλός βαθμός συσχέτισης και σειριακού χαρακτήρα που έχουν οι λειτουργίες αυτών των οντοτήτων μεταξύ τους, γεγονός που κάνει μη αναγκαίο τον παραλληλισμό τους. Ωστόσο κάποιες λειτουργίες που ενσωματώνονται στο task διαχείρισης συνδέσεων μπορεί να εκτελούνται και αυτόνομα (π.χ. η επεξεργασία ενός ICP\_PING πακέτου δεν απαιτεί την εκτέλεση της Ο.Δ.Σ.) γεγονός που μπορεί να

δικαιολογήσει την υλοποίηση του ICP σε ξεχωριστό task ώστε η αυτόνομη εκτέλεση να μην καθυστερεί από την εκτέλεση άλλων λειτουργιών (Ο.Δ.Σ.) οι οποίες μάλιστα λόγω χειρισμού φυσικών στοιχείων μπορεί και να μπλοκάρουν. Όμως η συχνότητα των αυτόνομων λειτουργιών (π.χ. ICP\_PING) σε σχέση με τον χρόνο εκτέλεσης και την πιθανότητα και καθυστέρηση του blocking των υπολοίπων λειτουργιών του task (Ο.Δ.Σ.) είναι τόσο μεγάλη που το κόστος για την υλοποίηση ενός επιπλέον task είναι κατά πολύ μεγαλύτερη από το κόστος καθυστέρησης των αυτόνομων υπολειτουργιών του task.

2. Ένα ξεχωριστό task αφιερώθηκε στην Ο.Ε.Δ.Σ. Ο λόγος για αυτό είναι προφανής, θέλουμε το task ελέγχου να εκτελείται ανεξάρτητα και συνήθως με την μέγιστη προτεραιότητα έτσι ώστε όταν ενεργοποιείται να εκτελείται υπερκαλύπτοντας την εκτέλεση των υπολοίπων (preemption) και να έχουμε με αυτόν τον τρόπο την ταχύτερη δυνατή απόκριση.
3. Τέλος σε ένα ξεχωριστό task εκτελείται ο αλγόριθμος ελέγχου ροής μαζί βεβαίως με της λειτουργίες RA που τον συνοδεύουν. Λόγω των υψηλών απαιτήσεων σε ταχύτητα εκτέλεσης και απόκριση (επεξεργάζεται cells που καταφθάνουν με υψηλούς ρυθμούς), το εν λόγω task εκτελείται εν παραλλήλω με τις υπόλοιπες οντότητες και με μικρότερη προτεραιότητα ώστε η συνεχής επεξεργασία των RM cells να μην μπλοκάρει τις υπόλοιπες διεργασίες. Στο task του ελέγχου ροής ενσωματώνεται και η P.X.Δ. που διαχειρίζεται την ανταλλαγή RM cells με το φυσικό στοιχείο. Αυτό έχει σημαντικά πλεονεκτήματα καθώς εξοικονομείται το κόστος ενός επιπλέον task σε λειτουργίες που είναι σειριακές. Αυτό το κόστος (κυρίως χρόνος context switch) μπορεί να αποβεί σημαντικό σχετιζόμενο με τις κλίμακες χρόνου και συχνότητας εκτέλεσης της διεργασίας ελέγχου ροής. Τέλος σε περίπτωση που το σύστημα ρυθμιστεί με τέτοιο τρόπο ώστε τα RM cells να εισάγονται από διαφορετικό φυσικό στοιχείο από ότι τα cells διαχείρισης τότε είναι δυνατόν να παραλειφθεί η επικοινωνία του task ελέγχου ροής μέσω διακοπών και να πραγματοποιείται μέσω polling. Αυτό μπορεί να εξοικονομήσει τον χρόνο που απαιτείται για την ανίχνευση των διακοπών και την εκτέλεση της ρουτίνας εξυπηρέτησης διακοπών.

Στην κάρτα εξυπηρετητή ABR σχεδιάστηκαν, υλοποιήθηκαν και ελέγχθηκαν τρεις διαφορετικές εκδόσεις του λογισμικού. Η λειτουργία αυτών των εκδόσεων, η οργάνωση τους σε tasks και η επικοινωνία μεταξύ των διαφόρων οντοτήτων περιγράφονται στις επόμενες τρεις παραγράφους.

### **3.3.8.2 Έκδοση με υποστήριξη μόνο λειτουργιών διαχείρισης**

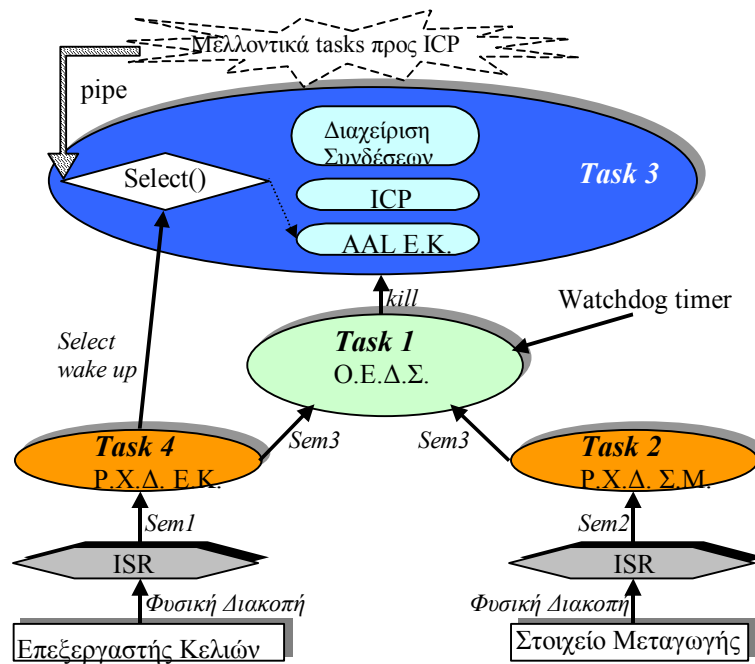
Η συγκεκριμένη έκδοση είναι η πιο «ελαφριά» και η ελάχιστη για να αποτελεί η κάρτα Εξυπηρετητή ABR λειτουργικό κομμάτι του Κόμβου Πρόσβασης. Η οργάνωση της φαίνεται στην Εικόνα 38.

Το λογισμικό σε αυτή την έκδοση αποτελείται από όλα τα tasks που περιγράφηκαν στην προηγούμενη παράγραφο εκτός από το task ελέγχου ροής. Οι αριθμοί που βρίσκονται δίπλα στα tasks δείχνουν την σχετική τους προτεραιότητα.

Δύο ρουτίνες εξυπηρέτησης διακοπών συνδέονται με τις φυσικές διακοπές του Επεξεργαστή Κελιών και του Στοιχείου Μεταγωγής αντίστοιχα. Αυτές οι ρουτίνες σε περίπτωση διακοπών τις ανιχνεύουν και ενεργοποιούν το task της αντίστοιχης P.X.Δ. (task2, task4) Η ενεργοποίηση αυτή πραγματοποιείται μέσω σηματοφόρων (Sem1, Sem2). Αν ενεργοποιηθεί η P.X.Δ. του Σ.Μ. τότε αυτή ελέγχει για πιθανά σφάλματα στο στοιχείο και αν βρεθούν ενημερώνει το task Ο.Ε.Δ.Σ. (task1) μέσω πάλι ενός σηματοφόρου (Sem3) που το ενεργοποιεί. Αν υπάρξει διακοπή στον Ε.Κ. και ενεργοποιηθεί η P.X.Δ. αυτού του στοιχείου, τότε ελέγχει και αυτή αν υπάρχουν σφάλματα και αν ναι ενημερώνει το task1 μέσω του σηματοφόρου με τον οποίο το τελευταίο ενεργοποιείται. Αν η διακοπή πραγματοποιείται για την ανταλλαγή cells διαχείρισης με τον επεξεργαστή τότε η P.X.Δ. του Ε.Κ. αποστέλλει ή λαμβάνει cells από το στοιχείο και εάν υπάρξει είσοδος για τον οδηγό AAL ενημερώνει το task διαχείρισης συνδέσεων.

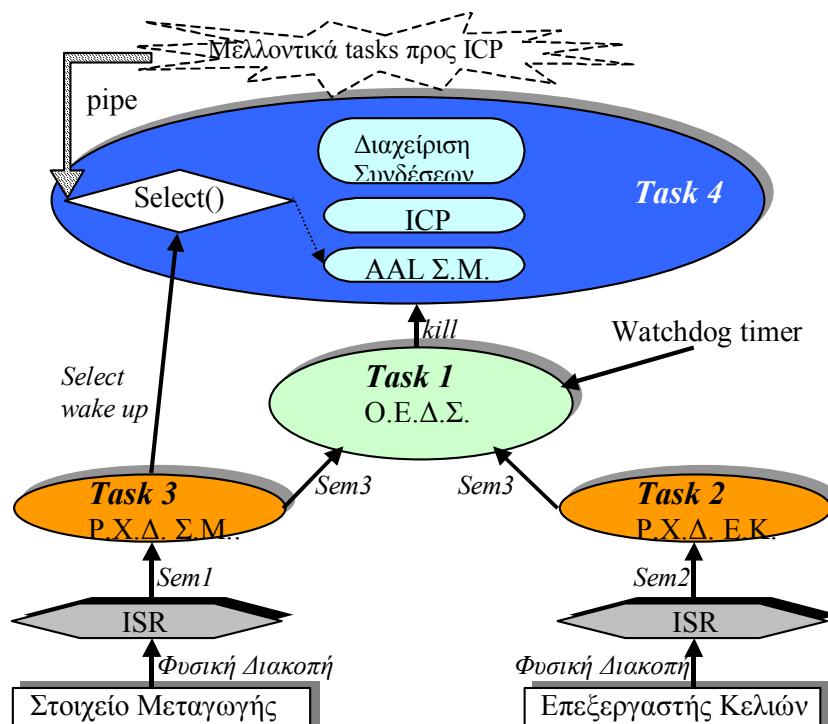
Το task διαχείρισης συνδέσεων (task3) υλοποιεί εσωτερικά ρουτίνα select() ή οποία περιμένει είσοδο σε κάποιον από τους οδηγούς με τους οποίους συνδέεται. Το task 3 συνδέεται με τον οδηγό AAL μέσω της αντίστοιχης P.X.Δ. ενώ επίσης εκκινεί και ένα pipe για την επικοινωνία του ICP με άλλους οδηγούς ή tasks εφαρμογών που μπορεί να ολοκληρωθούν με το σύστημα σε μία μεταγενέστερη έκδοση. Όταν η P.X.Δ. του Ε.Κ. θέλει να ενεργοποιήσει το task διαχείρισης συνδέσεων εκτελεί μία κλήση συστήματος select wakeur ώστε η ρουτίνα select του τελευταίου να αντιληφθεί ότι υπάρχει είσοδος για τον οδηγό AAL. Σε αυτή την περίπτωση στο context του task4 καλείται ο οδηγός AAL και στην συνέχεια εκτελούνται οι εργασίες του ICP και της διαχείρισης συνδέσεων σειριακά.

Παρατηρούμε ότι το task διαχείρισης συνδέσεων έχει μικρότερη προτεραιότητα έναντι του task Ο.Ε.Δ.Σ. για τους λόγους που αναφέρθηκαν στην παράγραφο 3.3.8.1. Επίσης η P.X.Δ. του Στοιχείου Μεταγωγής έχει μεγαλύτερη προτεραιότητα από αυτή του Επεξεργαστή Κελιών καθώς η πρώτη ενεργοποιείται μόνο σε περιπτώσεις σφάλματος οπότε η απόκριση πρέπει να είναι γρήγορη.



**Εικόνα 38 Οργάνωση Λογισμικού για την έκδοση που υποστηρίζει μόνο Διαχείριση (οδηγός AAL πάνω από E.K.)**

Η ίδια έκδοση μπορεί να υλοποιηθεί και με μία υποέκδοση όπου ο οδηγός AAL τρέχει πάνω από το Στοιχείο Μεταγωγής αντί του Επεξεργαστή Κελιών. Αυτή η υποέκδοση πρέπει να υποστηρίζεται και από την γενική λειτουργικότητα του Κόμβου Πρόσβασης. Σε αυτήν την περίπτωση η οργάνωση του λογισμικού είναι όμοια με την διαφορά ότι το task διαχείρισης συνδέσεων δέχεται είσοδο από το Σ.Μ. Οι αλλαγές φαίνονται στην Εικόνα 39.



Εικόνα 39 Οργάνωση Λογισμικού για την έκδοση που υποστηρίζει μόνο Διαχείριση (οδηγός AAL πάνω από Σ.Μ.)

### 3.3.8.3 Έκδοση με υποστήριξη λειτουργιών διαχείρισης και ελέγχου ροής πάνω από κοινό φυσικό στοιχείο (E.K.)

Σε αυτήν την έκδοση υποστηρίζονται τόσο οι λειτουργίες διαχείρισης όσο και οι λειτουργίες ελέγχου ροής. Τα cells που είναι απαραίτητα για την εκτέλεση αυτών των λειτουργιών (cells διαχείρισης και RM cells) εισάγονται/εξάγονται στον/από τον επεξεργαστή μέσω ενός κοινού φυσικού στοιχείου που είναι ο Επεξεργαστής Κελιών. Η οργάνωση αυτής της έκδοσης φαίνεται στην Εικόνα 40.

Το λογισμικό σε αυτή την έκδοση αποτελείται από όλα τα tasks που περιγράφηκαν στην προηγούμενη παράγραφο. Οι αριθμοί που βρίσκονται δίπλα στα tasks δείχνουν την σχετική τους προτεραιότητα.

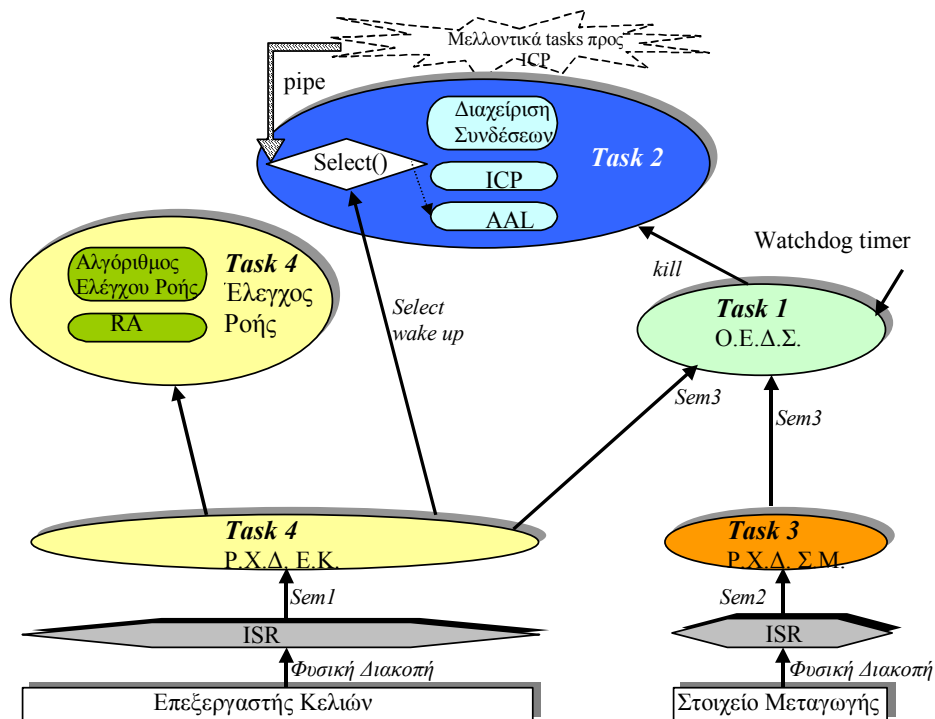
Δύο ρουτίνες εξυπηρέτησης διακοπών συνδέονται με τις φυσικές διακοπές του Επεξεργαστή Κελιών και του Στοιχείου Μεταγωγής αντίστοιχα. Αυτές οι ρουτίνες σε περίπτωση διακοπών, τις ανιχνεύουν και ενεργοποιούν το task της αντίστοιχης P.X.Δ. (task4, task5) Η ενεργοποίηση αυτή πραγματοποιείται μέσω σηματοφόρων (Sem1, Sem2). Αν ενεργοποιηθεί η P.X.Δ. του Σ.Μ. τότε αυτή ελέγχει για πιθανά σφάλματα στο στοιχείο και αν βρεθούν ενημερώνει το task O.E.Δ.Σ. (task1) μέσω πάλι ενός σηματοφόρου (Sem3) που το ενεργοποιεί.

Αν υπάρξει διακοπή στον E.K. και ενεργοποιηθεί η P.X.Δ. αυτού του στοιχείου, τότε ελέγχει και αυτή αν υπάρχουν σφάλματα και αν ναι ενημερώνει το task1 μέσω του σηματοφόρου με τον οποίο το τελευταίο ενεργοποιείται. Αν η διακοπή πραγματοποιείται για την ανταλλαγή cells με τον επεξεργαστή τότε η P.X.Δ. του E.K.



αποστέλει ή λαμβάνει cells από το στοιχείο. Στη συνέχεια ελέγχει το είδος των cells που εισήχθησαν. Αν πρόκειται για cells διαχείρισης τότε και εάν υπάρξει είσοδος για τον οδηγό AAL ενημερώνει το task διαχείρισης συνδέσεων και συγκεκριμένα ενεργοποιεί την διαδικασία ξυπνάει την ρουτίνα select (βλ 3.3.8.2). Αν πρόκειται για RM cells τότε η P.X.Δ. του E.K. εκτελεί την διαδικασία ελέγχου ροής στο ίδιο context. Αυτή διαβάζει τα εισερχόμενα RM cells τα επεξεργάζεται, τα ανανεώνει και τα αποστέλει πάλι στον E.K. μέσω της P.X.Δ.

Παρατηρούμε ότι το task ελέγχου ροής έχει μικρότερη προτεραιότητα έναντι του task διαχείρισης συνδέσεων και μικρότερη έναντι του task O.E.Δ.Σ. για τους λόγους που αναφέρθηκαν στην παράγραφο 3.3.8.1. Επίσης η P.X.Δ. του Στοιχείου Μεταγωγής έχει μεγαλύτερη προτεραιότητα από αυτή του Επεξεργαστή Κελιών καθώς η πρώτη ενεργοποιείται μόνο σε περιπτώσεις σφάλματος οπότε η απόκριση πρέπει να είναι γρήγορη.



Εικόνα 40 Οργάνωση Λογισμικού για την έκδοση που υποστηρίζει Διαχείριση και Έλεγχο Ροής με χειρισμό cells από το ίδιο φυσικό στοιχείο

### 3.3.8.4 Έκδοση με υποστήριξη λειτουργιών διαχείρισης και ελέγχου ροής πάνω από διαφορετικά φυσικά στοιχεία (Σ.Μ και E.K.)

Σε αυτή την έκδοση ο χειρισμός των εισερχόμενων cells από τις οντότητες υψηλότερου επιπέδου είναι κατανομημένος. Συγκεκριμένα τα cells διαχείρισης ανταλλάσσονται μεταξύ επεξεργαστή και Σ.Μ. ενώ τα RM cells ανταλλάσσονται μεταξύ επεξεργαστή και E.K. Η κατανομημένη λειτουργία αυτής της έκδοσης την καθιστά περισσότερο αποτελεσματική και πιο εύκολα οργανώσιμη. Παρόλα αυτά στην εφαρμογή της υπάρχουν ορισμένα εμπόδια καθώς απαιτείται ο οδηγός AAL να είναι

υλοποιημένος να τρέχει πάνω από το Σ.Μ. και επίσης θα πρέπει το κεντρικό λογισμικό του Κόμβου Πρόσβασης να υποστηρίζει ανταλλαγή cells διαχείρισης μέσω του Σ.Μ. Η διαμετρική υλοποίηση (cells διαχείρισης μέσω Ε.Κ. / RM cells μέσω Σ.Μ.) δεν είναι εφικτή καθώς το Σ.Μ. δεν υποστηρίζει ανταλλαγή RM cells με τον επεξεργαστή.

Η οργάνωση αυτής της έκδοσης φαίνεται στην Εικόνα 41.

Το λογισμικό σε αυτή την έκδοση αποτελείται από όλα τα tasks που περιγράφηκαν στην προηγούμενη παράγραφο και από το task ελέγχου ροής. Οι αριθμοί που βρίσκονται δίπλα στα tasks δείχνουν την σχετική τους προτεραιότητα.

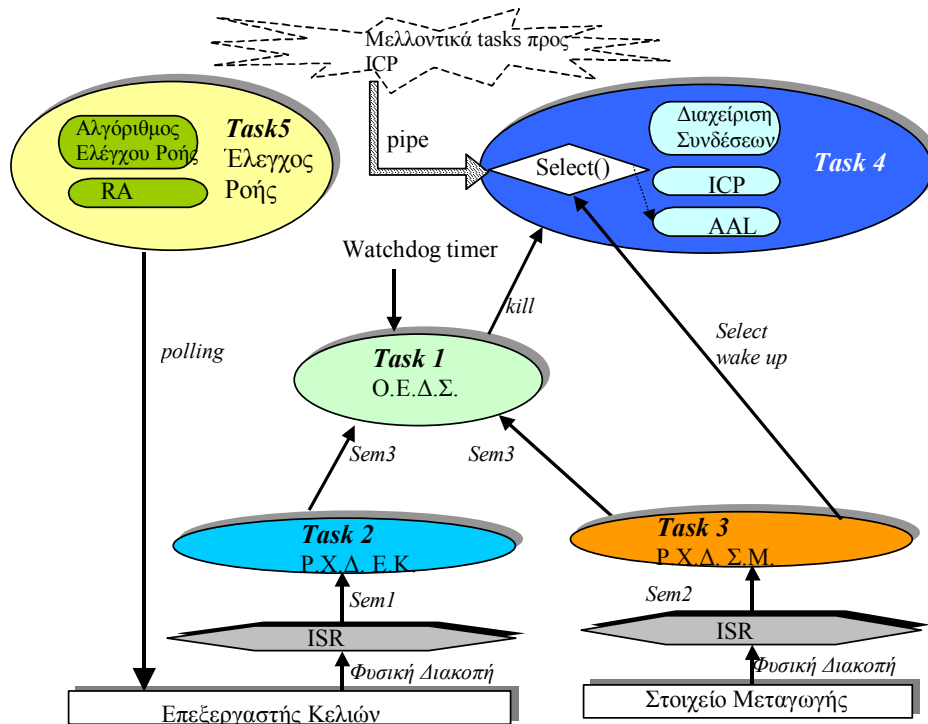
Δύο ρουτίνες εξυπηρέτησης διακοπών συνδέονται με τις φυσικές διακοπές του Επεξεργαστή Κελιών και του Στοιχείου Μεταγωγής αντίστοιχα. Αυτές οι ρουτίνες σε περίπτωση διακοπών, τις ανιχνεύουν και ενεργοποιούν το task της αντίστοιχης Ρ.Χ.Δ. (task2, task3) Η ενεργοποίηση αυτή πραγματοποιείται μέσω σηματοφόρων (Sem1, Sem2).

Αν ενεργοποιηθεί η Ρ.Χ.Δ. του Σ.Μ. τότε αυτή ελέγχει για πιθανά σφάλματα στο στοιχείο και αν βρεθούν ενημερώνει το task Ο.Ε.Δ.Σ. (task1) μέσω πάλι ενός σηματοφόρου (Sem3) που το ενεργοποιεί. Αν η διακοπή πραγματοποιείται για την ανταλλαγή cells με τον επεξεργαστή τότε η Ρ.Χ.Δ. του Σ.Μ. αποστέλλει ή λαμβάνει cells διαχείρισης από το στοιχείο. Εάν υπάρξει είσοδος για τον οδηγό AAL ενημερώνει το task διαχείρισης συνδέσεων και συγκεκριμένα ενεργοποιεί την διαδικασία ξυπνώντας την ρουτίνα select (βλ 3.3.8.2).

Αν υπάρξει διακοπή στον Ε.Κ. και ενεργοποιηθεί η Ρ.Χ.Δ. αυτού του στοιχείου, τότε ελέγχει αυτή αν υπάρχουν σφάλματα και αν ναι ενημερώνει το task1 μέσω του σηματοφόρου με τον οποίο το τελευταίο ενεργοποιείται.

Το task ελέγχου ροής επικοινωνεί με τον Ε.Κ. κάνοντας polling. Όταν αντιληφθεί ότι υπάρχει σε αυτόν RM cells τότε διαβάζει τα εισερχόμενα RM cells τα επεξεργάζεται, τα ανανεώνει και τα αποστέλλει πάλι στον Ε.Κ. όταν αυτός είναι έτοιμος να δεχθεί cells. Το τελευταίο γίνεται αντιληπτό στο task ελέγχου ροής ξανά μέσω polling. Το συγκεκριμένο task δεν επικοινωνεί με τον Ε.Κ. μέσω διακοπών για να εξοικονομηθεί ο χρόνος που σπαταλάται στην εξυπηρέτηση των διακοπών.

Παρατηρούμε ότι το task ελέγχου ροής έχει μικρότερη προτεραιότητα ενάντι του task διαχείρισης συνδέσεων και του task Ο.Ε.Δ.Σ. για τους λόγους που αναφέρθηκαν στην παράγραφο 3.3.8.1.



Εικόνα 41 Οργάνωση Λογισμικού για την έκδοση που υποστηρίζει Διαχείριση και Έλεγχο Ροής με χειρισμό cells από διαφορετικά φυσικά στοιχεία

### 3.4 Υλοποίηση λογισμικού

#### 3.4.1 Περιγραφή κώδικα

Το λογισμικό της κάρτας Εξυπηρετητή ABR υλοποιήθηκε σε γλώσσα C και αποτελείται από 8 αρχεία τα οποία περιέχουν τον κώδικα που υλοποιεί τις διάφορες λειτουργίες. Η υλοποίηση πραγματοποιήθηκε με τέτοιο τρόπο ώστε κάθε αρχείο να αντιστοιχεί σε ένα στρώμα της ιεραρχίας του λογισμικού.

Ένα μέρος του κώδικα υλοποιήθηκε από το Ε.Μ.Π. και ενσωματώθηκε ως είχε στον κώδικα που υλοποιήθηκε σε αυτή την εργασία. Ένα δεύτερο μέρος του κώδικα υλοποιήθηκε επίσης από το Ε.Μ.Π. αλλά μετατράπηκε πλήρως και ενσωματώθηκε στον τελικό κώδικα. Τέλος το μεγαλύτερο μέρος του κώδικα υλοποιήθηκε εξαρχής στα πλαίσια της παρούσας εργασίας. Τα αρχεία που χρησιμοποιήθηκαν και ο φορέας υλοποίησης τους είναι τα εξής:

- *MC92501.c* Υλοποιεί την Δ.Χ.Φ.Σ. και τις ρουτίνες προγραμματισμού του Επεξεργαστή Κελιών(MC92501) - Υλοποιήθηκε αρχικά από το Ε.Μ.Π και μετατράπηκε για να προσαρμοστεί στις ανάγκες της κάρτας Εξυπηρετητή ABR.
- *Fpga.c* Υλοποιεί την Δ.Χ.Φ.Σ. και τις ρουτίνες αρχικοποίησης της Μ.Ε.Κ.Α. (FPGA) – Υλοποιήθηκε εξ αρχής στα πλαίσια της παρούσας εργασίας

- *Atm.c*                    **Υλοποιεί τις λειτουργίες του επιπέδου ATM** - Υλοποιήθηκε από το Ε.Μ.Π.
- *AALDriver.c*           **Υλοποιεί τον οδηγό AAL** - Υλοποιήθηκε από το Ε.Μ.Π και προσαρμόστηκε στις εφαρμογές τις παρούσας εργασίας.
- *Icp.c*                    **Υλοποιεί τις λειτουργίες του ICP** - Υλοποιήθηκε εξ αρχής στα πλαίσια της παρούσας εργασίας
- *Management.c*       **Υλοποιεί τις λειτουργίες του επιπέδου διαχείρισης** - Υλοποιήθηκε εξ αρχής στα πλαίσια της παρούσας εργασίας
- *Rmcells.c*            **Υλοποιεί τις λειτουργίες του επιπέδου RA και της εφαρμογής Ελέγχου Ροής** - Υλοποιήθηκε εξ αρχής στα πλαίσια της παρούσας εργασίας

Η πλήρης περιγραφή της υλοποίησης (κώδικας) βρίσκεται στο **Error! Reference source not found.**

Ο συνολικός κώδικας αποτελείται από 2800 γραμμές κώδικα C. Από αυτές οι 1450 (51%) υλοποιήθηκαν εξ αρχής, οι 550 (20%) μετατράπηκαν από έτοιμο κώδικα και οι 800 (29%) χρησιμοποιήθηκαν ως είχαν και προσαρμόστηκαν στον υπόλοιπο κώδικα.

### 3.4.2 Εργαλεία

Για την ανάπτυξη και τον έλεγχο του λογισμικού χρησιμοποιήθηκαν τα εργαλεία της εταιρίας AG Electronics. Συγκεκριμένα χρησιμοποιήθηκαν:

- AG Gnu Compiler – Compiler για τον Επεξεργαστή MPC860SAR της Motorola
- AG Standard C Lib – Πρότυπη βιβλιοθήκη της C για τον Επεξεργαστή MPC860SAR
- AG Gnu GDB – Έκδοση του γνωστού debugger για τον MPC860SAR.

### 3.4.3 Εγκατάσταση

Με την χρήση των εργαλείων δημιουργήθηκε ένα .out αρχείο σε μορφή .ELF. Αυτό εγκαταστάθηκε στο λειτουργικό της κάρτας. Η μέθοδος εγκατάστασης και ελέγχου του λειτουργικού αναφέρεται αναλυτικά στο Κεφάλαιο 4

Με την εκκίνηση του συστήματος, αυτό είτε αυτόματα (boot system) είτε με εξωτερικό έλεγχο εκκινεί το λογισμικό. Η εκκίνηση ακολουθεί τα παρακάτω βήματα:

1. Αμέσως μετά την εκκίνηση του συστήματος, την αρχικοποίηση του επεξεργαστή και την εκκίνηση του λειτουργικού καλείται η συνάρτηση ABR\_Server\_Init()
2. Η ABR\_Server\_Init() καλεί με την σειρά της τις συναρτήσεις,
  - encb() – εγκαθιστά το device Cubit
  - encp() – εγκαθιστά το device Cell Processor

- enfpga() – εγκαθιστά το device FPGA
- initDrv() – εγκαθιστά τον Οδηγό AAL – Προγραμματίζει και αρχικοποιεί το φυσικό στοιχείο πάνω από το οποίο τρέχει.
- ICPCClient\_Init() – Αρχικοποιεί το task ICP
- InitManagement() – Αρχικοποιεί τα tasks της Διαχείρισης – Προγραμματίζει και αρχικοποιεί τα υπόλοιπα φυσικά στοιχεία.
- InitRMcellMarking() – Αρχικοποιεί τα tasks για τον έλεγχο ροής (εφόσον το υποστηρίζει η έκδοση) – Προγραμματίζει και Αρχικοποιεί το φυσικό στοιχείο πάνω από το οποίο τρέχει η εφαρμογή ελέγχου ροής

## 4 Εγκατάσταση, Έλεγχος και Μετρήσεις του συστήματος

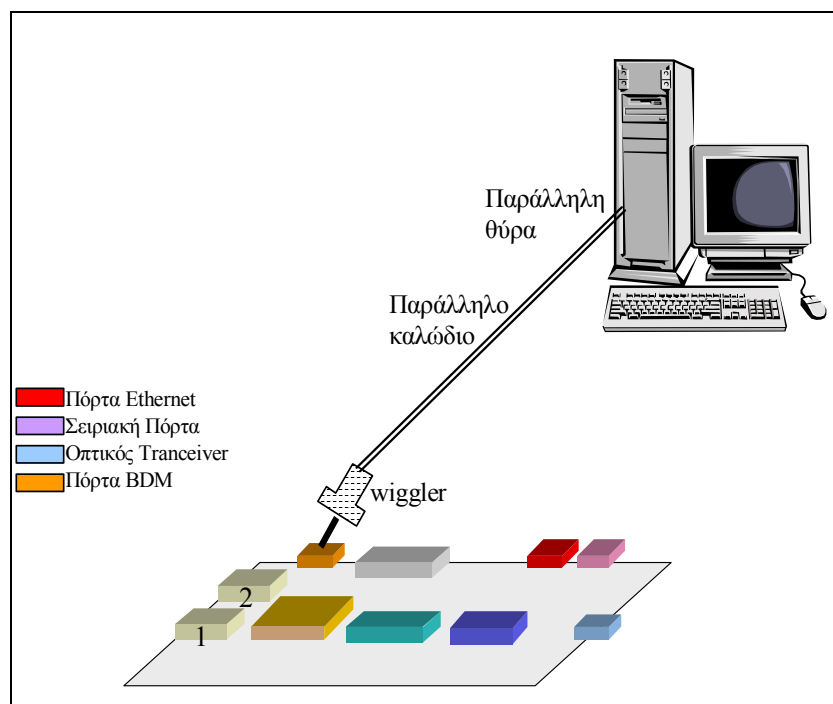
Σε αυτό το κεφάλαιο θα περιγραφούν οι διατάξεις και τα βήματα που ακολουθήθηκαν για τον έλεγχο της λειτουργικότητας της κάρτας εξυπηρετητή ABR, την εγκατάσταση του λογισμικού σε αυτήν καθώς και για τον έλεγχο της σωστής λειτουργίας του. Επίσης θα παρατεθούν και στοιχεία από μετρήσεις για την απόδοση των οντοτήτων λογισμικού που υλοποιήθηκαν.

### 4.1 Έλεγχος του Υλικού της κάρτας

Το πρώτο βήμα στην διαδικασία ελέγχου του συστήματος ήταν ο έλεγχος του υλικού της κάρτας. Σκοπός ήταν να εξακριβωθεί ότι τα φυσικά στοιχεία της κάρτας είναι ενεργά και προγραμματίσιμα από τον επεξεργαστή. Συγκεκριμένα ελέγχθηκε η σωστή λειτουργία των διεπαφών των φυσικών στοιχείων με τον επεξεργαστή καθώς και η φυσική λειτουργία του διαύλου πάνω από τον οποίο πραγματοποιείται η επικοινωνία τους. Επίσης ελέγχθηκε η δυνατότητα αρχικοποίησης και η ορθή λειτουργικότητα τους με βάση τον προγραμματισμό τους.

Η διάταξη που χρησιμοποιήθηκε για τον έλεγχο του υλικού της κάρτας βασίστηκε σε μία δυνατότητα άμεσου εσωτερικού ελέγχου του επεξεργαστή MPC860. Αυτός μπορεί να πραγματοποιηθεί μέσω της πόρτας BDM (Background Debugging Mode) με την οποία είναι εξοπλισμένος ο επεξεργαστής [8]. Μέσω αυτής της πόρτας είναι δυνατός ο έλεγχος, η αρχικοποίηση, ο εσωτερικός προγραμματισμός και ο χειρισμός των διεπαφών του επεξεργαστή με τον έξω κόσμο, από ένα εξωτερικό υπολογιστή ελέγχου. Μέσω του BDM ο επεξεργαστής μπορεί να δεχθεί εντολές χαμηλού επιπέδου από ένα εξωτερικό μηχάνημα. Κατά την επικοινωνία με το BDM ο επεξεργαστής σταματά την κανονική λειτουργία του και μπαίνει σε κατάσταση που ονομάζεται debugging mode στην οποία δέχεται εντολές από την πόρτα BDM και όχι από την instruction cache του.

Για την επικοινωνία με την πόρτα BDM του MPC860 χρησιμοποιήθηκε μία ειδική συσκευή που ονομάζεται Wiggler και κατασκευάζεται από την εταιρία McGraigor Systems. Η διάταξη που χρησιμοποιήθηκε φαίνεται στην Εικόνα 42.



**Εικόνα 42 Διάταξη για επικοινωνία του Host με την πόρτα BDM του επεξεργαστή.**

Ως σύστημα ελέγχου χρησιμοποιήθηκε ένα PC. Αυτό είναι συνδεδεμένο με την κάρτα μέσω της παράλληλης πόρτας του με ένα παράλληλο καλώδιο στο οποίο είναι συνδεδεμένο το εξάρτημα Wiggler. Αυτό με την σειρά του είναι συνδεδεμένο στην πόρτα BDM της κάρτας και του επεξεργαστή. Στο host PC τρέχει κατάλληλο λογισμικό που συνοδεύει το εξάρτημα Wiggler (OCD Commander της McGraigor Systems) και μέσω του οποίου ελέγχεται η εσωτερική λειτουργία του MPC860.

Η διαδικασία ελέγχου που ακολουθήθηκε είναι η εξής:

1. Αρχικοποιήθηκαν οι εσωτερικοί καταχωρητές του MPC860 ώστε:
  - a. Να καθοριστεί η δομή του διαύλου επικοινωνίας με τα φυσικά στοιχεία της κάρτας
  - b. Να καθοριστεί η συχνότητα των ρολογιών που παρέχει ο επεξεργαστής στα φυσικά στοιχεία της κάρτας
  - c. Να προγραμματιστούν οι καταχωρητές που ρυθμίζουν την αντιστοίχιση διεθύνσεων για κάθε φυσικό στοιχείο καθώς και τις παραμέτρους επικοινωνίας του επεξεργαστή με τα φυσικά στοιχεία
2. Αρχικοποιήθηκαν τα φυσικά στοιχεία πραγματοποιώντας reset μέσω του επεξεργαστή σε κάθε ένα από αυτά

3. Αναγνώστηκαν οι εσωτερικοί καταχωρητές του κάθε στοιχείου για να ελεγχθούν οι reset τιμές καθώς και η ορθή λειτουργία της διαδικασίας ανάγνωσης των φυσικών στοιχείων μέσω του διαύλου
4. Εγγράφηκαν καταχωρητές διεπαφής με τον επεξεργαστή των φυσικών στοιχείων με συγκεκριμένες τιμές για να ελεγχθεί η ορθή λειτουργία της διαδικασίας εγγραφής μέσω του διαύλου.
5. Πραγματοποιήθηκε ένας αρχικός προγραμματισμός των εμπορικών στοιχείων για να ελεγχθεί η ορθή λειτουργία τους
6. Πραγματοποιήθηκε ένας αρχικός προγραμματισμός και διαδικασία παροχής εντολών στην F.P.G.A. που υλοποιεί την M.E.K.A. για να ελεγχθεί η ορθή λειτουργία της διεπαφής με τον επεξεργαστή.

## 4.2 Εγκατάσταση Λογισμικού

Σε αυτή την παράγραφο περιγράφεται η διαδικασία εγκατάστασης του υλοποιημένου λογισμικού στην Κάρτα Εξυπηρετητή ABR. Αναφέρονται τα βήματα εγκατάστασης για τον έλεγχο του λογισμικού καθώς και για την μόνιμη εγκατάσταση σε ένα λειτουργιών σύστημα.

### 4.2.1 Εγκατάσταση λειτουργικού συστήματος

Το πρώτο βήμα στην διαδικασία εγκατάστασης του λογισμικού, είναι η εγκατάσταση του λειτουργικού συστήματος. Η εγκατάσταση αυτή πρέπει να γίνει πρώτη καθώς η επιμέρους εγκατάσταση των υπόλοιπων οντοτήτων λογισμικού βασίζεται σε λειτουργικότητα που παρέχει το λειτουργικό σύστημα.

Η μόνιμη εγκατάσταση του λειτουργικού συστήματος στην κάρτα πραγματοποιείται μέσω της αποθήκευσης ενός image του συστήματος στις μνήμες Flash EPROM. Κατά την εκκίνηση του συστήματος ο επεξεργαστής αυτομάτως ξεκινά την εκτέλεση του κώδικα που βρίσκεται στις συγκεκριμένες μνήμες. Ο κώδικας αυτός αρχικά αντιγράφει τον κώδικα του λειτουργικού από τις Flash στην κύρια μνήμη (SDRAM) και στην συνέχεια εκκινεί την λειτουργία μεταφέροντας τον program counter στην θέση μνήμης από την οποία ξεκινά ο κώδικας του λειτουργικού.

Η διαδικασία downloading του image του λειτουργικού της κάρτας στις Flash EPROMs πραγματοποιείται με την βοήθεια της διάταξης της **Error! Reference source not found.** και έχει ως εξής:

Το λογισμικό που χρησιμοποιείται για την επικοινωνία με την πόρτα BDM υποστηρίζει διαδικασία downloading ενός αρχείου στην μνήμη SDRAM του επεξεργαστή. Μέσω αυτής της διαδικασίας αρχικά φορτώνεται το image file του λειτουργικού συστήματος στην μνήμη. Στην συνέχεια φορτώνεται στην μνήμη ένας μικρός assembly κώδικας ο οποίος μπορεί να αντιγράψει τα περιεχόμενα της μνήμης στα οποία βρίσκεται το λειτουργικό στην Flash EPROM. Τελικά πάλι μέσω του λογισμικού της πόρτας BDM δίδεται εντολή στον επεξεργαστή να ξεκινήσει να εκτελεί εντολές από την πρώτη θέση μνήμης του προγράμματος αντιγραφής, δηλαδή

να ξεκινήσει να εκτελεί το συγκεκριμένο πρόγραμμα. Με την παρέλευση μικρού χρονικού διαστήματος ο κώδικας του λειτουργικού έχει φορτωθεί στις Flash EPROMs και έχει εγκατασταθεί μόνιμα στο σύστημα.

Από αυτή τη στιγμή και μετά κάθε φορά που η κάρτα εκκινείται (power up) το λειτουργικό σύστημα ξεκινά αυτόματα κάνοντας αυτό όλες τις απαραίτητες διαδικασίες για την σωστή λειτουργία του επεξεργαστή. Η πόρτα BDM πλέον δεν είναι απαραίτητη καθώς ο επεξεργαστής είναι υπό τον έλεγχο του λειτουργικού και οι λειτουργίες της κάρτας καθορίζονται μέσω των δυνατοτήτων που παρέχονται από αυτό.

Η επικοινωνία του χρήστη με το λειτουργικό και τον επεξεργαστή γίνεται μέσω του κελύφους (shell) που αυτό παρέχει. Το λειτουργικό θεωρεί ως μονάδα εξόδου του την σειριακή θύρα του MPC860. Έτσι συνδέοντας την σειριακή πόρτα ενός εξωτερικού PC με την σειριακή πόρτα της κάρτας, χρησιμοποιείται το κέλυφος και οι δυνατότητες του [10] μέσω ενός hyperterminal.

#### 4.2.2 Εγκατάσταση εφαρμογών λογισμικού

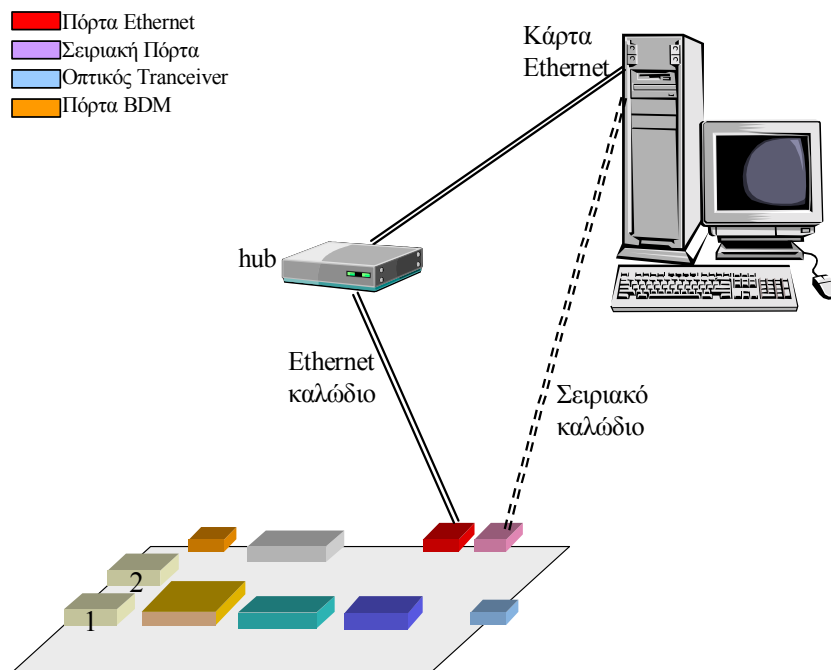
Η εγκατάσταση των εφαρμογών που υλοποιήθηκαν για να τρέχουν στον επεξεργαστή της κάρτας πραγματοποιείται μέσω λειτουργιών που παρέχει το λειτουργικό σύστημα και ελέγχονται από το κέλυφος του.

Αρχικά ο κώδικας που έχει γραφεί για τις εφαρμογές γίνεται compiled σε ένα εξωτερικό host ώστε να παραχθούν τα αντικειμενικά αρχεία (object files) των εφαρμογών. Τα αντικειμενικά αρχεία πρέπει να είναι τύπου .ELF ώστε να υποστηρίζονται από το λειτουργικό σύστημα. Στη συνέχεια τα αντικειμενικά αρχεία γίνονται download στον επεξεργαστή και στο λειτουργικό.

Τα αντικειμενικά αρχεία αφού «κατέβουν» στο λειτουργικό σύστημα είναι αμέσως εκτελέσιμα. Δεν απαιτείται να πραγματοποιηθεί linking μεταξύ των αντικειμενικών αρχείων και των βιβλιοθηκών του λειτουργικού εξωτερικά (όπως είναι το σύνηθες) πριν το download. Αυτό γιατί κατά την διάρκεια του download, το λειτουργικό έχει την δυνατότητα να πραγματοποιεί dynamic linking των αντικειμενικών αρχείων που κατεβαίνουν στο σύστημα με τα ήδη εγκατεστημένα προγράμματα. Για να πραγματοποιηθεί αυτό, το λειτουργικό σύστημα διατηρεί ένα εσωτερικό symbol table. Μέσω αυτού του symbol table είναι δυνατή η εκτέλεση μέσω του κελύφους οποιασδήποτε συνάρτησης C που είναι υλοποιημένη σε ένα downloaded αντικειμενικό αρχείο [10].

Η διαδικασία download των αντικειμενικών αρχείων στον επεξεργαστή πραγματοποιείται μέσω της διάταξης της **Error! Reference source not found.** και περιγράφεται στην συνέχεια:





Εικόνα 43 Διάταξη για την εγκατάσταση λογισμικού

Το λειτουργικό σύστημα περιέχει ενσωματωμένο έναν οδηγό Ethernet καθώς και έναν FTP client. Επίσης είναι ρυθμισμένο ώστε να κάνει download ενός αντικειμενικού αρχείου μέσω αυτού του FTP client. Στην Ethernet πόρτα της κάρτας συνδέεται μέσω κατάλληλου καλωδίου η Ethernet κάρτα του εξωτερικού PC ώστε να υπάρξει Ethernet σύνδεση μεταξύ κάρτας και PC. Στο εξωτερικό PC τρέχει ένας κατάλληλος FTP server και τα αντικειμενικά αρχεία τοποθετούνται στον αρχικό φάκελο του server.

Με αυτή τη διάταξη το λειτουργικό της κάρτας μπορεί να εγκαταστήσει μία FTP σύνδεση με το PC, αφού το πρώτο έχει ρυθμιστεί έτσι ώστε να γνωρίζει την διεύθυνση IP του δεύτερου και να γνωρίζει ένα login name και έναν κωδικό πρόσβασης στον FTP server.

Τελικά χρησιμοποιώντας μία εντολή *ld <αντικειμενικό αρχείο>* στο κέλυφος του λειτουργικού αυτό εκκινεί την FTP σύνδεση με το εξωτερικό PC, μεταφέρει το αντικειμενικό αρχείο στην μνήμη της κάρτας, πραγματοποιεί dynamic linking και ολοκληρώνει την εγκατάσταση του κώδικα στο σύστημα [10].

Με την παραπάνω διαδικασία εγκαθίσταται το λογισμικό για να ελεγχθεί η σωστή λειτουργία του πάνω στην κάρτα (debugging). Δεν είναι απαραίτητο να εγκαθιστάται ολόκληρο το λογισμικό κάθε φορά αλλά μόνο οι οντότητες που πρέπει να ελεγχθούν.

Όταν ο έλεγχος του λογισμικού ολοκληρωθεί και αυτό καταστεί πλήρως λειτουργικό, τότε είναι δυνατόν να εγκατασταθεί μόνιμα στο σύστημα αποθηκευόμενο μαζί με το λειτουργικό στις Flash EPROMS με την διαδικασία που περιγράφηκε στην παράγραφο 4.2.1. Σε αυτήν την περίπτωση κατά την εκκίνηση του συστήματος, εκκινείται πρώτα το λειτουργικό σύστημα και στην συνέχεια αυτό αναλαμβάνει να

εκτελέσει αυτόματα την ακολουθία εκκίνησης των εφαρμογών [10] ώστε να θέσει το σύστημα σε πλήρη λειτουργία.

### 4.3 Έλεγχος Λογισμικού

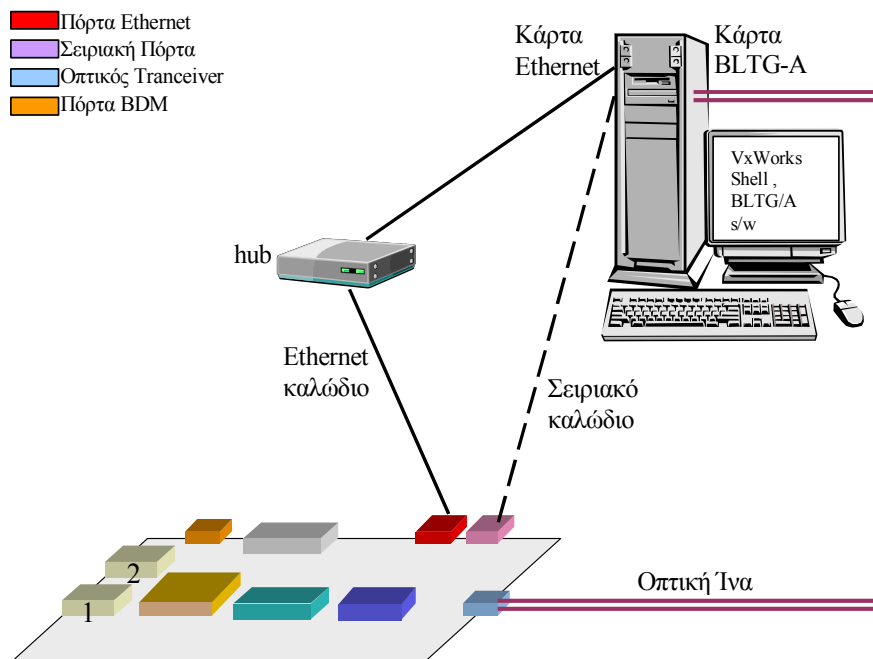
Ο έλεγχος του λογισμικού πραγματοποιήθηκε μέσω των εργαλείων που παρέχονται από το κέλυφος του λειτουργικού. Οι οντότητες του λογισμικού ελέγχθηκαν αρχικά επιμέρους και στην συνέχεια ολοκληρώθηκαν μεταξύ τους και ελέγχθηκε η συνολική λειτουργία του λογισμικού.

Κατά τον επιμέρους έλεγχο τα αντικειμενικά αρχεία που περιείχαν τον ελεγκτέο κώδικα εγκαταστάθηκαν σταδιακά στο λειτουργικό με την διαδικασία που περιγράφηκε προηγουμένως. Από το κέλυφος του λειτουργικού εκτελέστηκαν οι συναρτήσεις που θέτουν σε λειτουργία τον ελεγκτέο κώδικα καθώς και μεμονωμένες συναρτήσεις για να ελεγχθούν μικρότερα μέρη του κώδικα. Η διαδικασία ελέγχου ξεκίνησε από τα χαμηλά επίπεδα του λογισμικού και προχώρησε σταδιακά μέχρι τα υψηλότερα επίπεδα οπότε και ελέγχθηκαν ολοκληρωμένα οι εφαρμογές σε πλήρη λειτουργία. Η διαδικασία που ακολουθήθηκε καθώς και οι διατάξεις που χρησιμοποιήθηκαν αναφέρονται στην συνέχεια.

1. Αρχικά ελέγχθηκαν οι ρουτίνες προγραμματισμού των φυσικών στοιχείων. Οι ρουτίνες εκτελέστηκαν μέσω του κελύφους του λειτουργικού και η σωστή εγγραφή των τιμών των εσωτερικών καταχωρητών ελέγχθηκε μέσω της πόρτας BDM και του λογισμικού του Wiggler.
2. Στη συνέχεια ελέγχθηκαν οι Διεπαφές Χειρισμού Φυσικών Στοιχείων καθώς και οι ρουτίνες που υλοποιούν το επίπεδο ATM (λειτουργίες αποθήκευσης cells).
  - a. Πρώτα ελέγχθηκε η Διεπαφή Χειρισμού της M.E.K.A. και οι συναρτήσεις που σχηματίζουν και μεταφέρουν εντολές μεταξύ επεξεργαστή και M.E.K.A. Η ορθή μεταφορά των εντολών ελέγχθηκε διαβάζοντας τις τιμές στους καταχωρητές διεπαφής μικροεπεξεργαστή της M.E.K.A. μέσω της πόρτας BDM.
  - b. Στη συνέχεια ελέγχθηκε η Διεπαφή Χειρισμού του Επεξεργαστή Κελιών και συγκεκριμένα οι ρουτίνες για την εισαγωγή/εξαγωγή cells. Για αυτό το λόγο χρησιμοποιήθηκε μία γεννήτρια cells που διαθέτει το εργαστήριο Αρχιτεκτονικής Υπολογιστών. Το σύστημα αυτό ονομάζεται BLTG-A (της εταιρίας ForePlus) και τοποθετείται στο εξωτερικό PC ελέγχου μέσω μίας κάρτας. Επίσης διατίθεται το λογισμικό που ελέγχει την κάρτα και καθορίζει την παραγωγή, αποστολή και αποδοχή cells. Η κάρτα BLTG-A συνδέεται στο Physical chip της κάρτας Εξυπηρετητή ABR μέσω οπτικής ίνας οπότε το PC ελέγχου μπορεί να παρέχει κίνηση ATM στην κάρτα. Η κίνηση αυτή μπορεί να είναι οποιοδήποτε τύπου και ρυθμού μετάδοσης. Ο μέγιστος ρυθμός μετάδοσης είναι 155Mbps. Η διάταξη που χρησιμοποιήθηκε φαίνεται στην Εικόνα 44. Μέσω αυτής της διάταξης

στάλθηκαν cells στην κάρτα και ελέγχθηκε η εξαγωγή τους από τον Επεξεργαστή Κελιών προς τον επεξεργαστή. Ο Ε.Κ. προγραμματίστηκε να εξάγει όλα τα cells με συγκεκριμένο VP/VC και ο BLTG-A απέστειλε προς την κάρτα κίνηση με το ίδιο VP/VC. Ο Ε.Κ. απέστειλε τα cells προς την ουρά εξόδου και δημιούργησε την απαραίτητη διακοπή. Στην συνέχεια εκτελέστηκαν μέσω του κελύφους οι συναρτήσεις ανάγνωσης cell και αποθήκευσης στην ουρά εισόδου. Μέσω μηνυμάτων που εκτυπωνόταν στο κέλυφος (printfs) ελέγχθηκε η σωστή ανάγνωση των cells και η τοποθέτηση τους στην ουρά. Στη συνέχεια εκτελέστηκαν οι συναρτήσεις εξαγωγής από την ουρά και αποστολής των cells στον Ε.Κ. Ο Ε.Κ. απέστειλε τα cells που δέχθηκε προς την έξοδο της κάρτας και η ορθή λειτουργία ελέγχθηκε με την αποδοχή των cells από τον cell Analyzer της κάρτας BLTG-A.

3. Το επόμενο βήμα ήταν ο έλεγχος του οδηγού AAL. Πρώτα ελέγχθηκε η Ρουτίνα Χειρισμού Διακοπών του οδηγού χρησιμοποιώντας την διάταξη της Εικόνας 4.3. Μέσω του BLTG-A έφτασαν στον επεξεργαστή cells με PTI = 1 (last packet cell). Ελέγχθηκε η λειτουργία της PΧΔ, και η ενεργοποίηση των σηματοφόρων όταν ανιχνευθεί ένα πακέτο στο εσωτερικό του επεξεργαστή
4. Τέλος ελέγχθηκαν οι συναρτήσεις του Επιπέδου RA. Αυτό πραγματοποιήθηκε μέσω της διάταξης της Εικόνα 4.4. Ο BLTG-A παρήγαγε και απέστειλε στην κάρτα ABR κίνηση. Ο Ε.Κ. προγραμματίστηκε να μεταφέρει στον επεξεργαστή κελιών τα RM cells αυτής της κίνησης. Τα RM cells μεταφέρθηκαν και αποθηκεύθηκαν στο εσωτερικό του επεξεργαστή. Έπειτα εκτελέστηκαν οι ρουτίνες που εξάγουν τις παραμέτρους από τα RM cell καθώς και αυτές που πραγματοποιούν την μετατροπή της παράστασης των ρυθμών από αυτή που χρησιμοποιήτε στα RM cells σε αυτή που χρησιμοποιεί το λογισμικό. Μέσω μηνυμάτων στην οθόνη (printfs) εξακριβώθηκε η σωστή ανάγνωση και μετατροπή. Οι παράμετροι που διαβάστηκαν ήταν αυτοί που πραγματικά θα έπρεπε να μεταφέρουν τα RM cells σύμφωνα με το ρυθμό μετάδοσης τους ο οποίος ήταν προκαθορισμένος.



Εικόνα 44 Διάταξη για τον έλεγχο του λογισμικού με πραγματικά cells

Μετά τον επιμέρους έλεγχο των λειτουργιών που έχουν να κάνουν με το υλικό της κάρτας(χαμηλό επίπεδο), αυτές ολοκληρώθηκαν με της οντότητες υψηλότερου επιπέδου και ελέγχθηκε συνολικά η λειτουργία του λογισμικού ως εξής:

1. Ελέγχθηκαν οι λειτουργίες Διαχείρισης, ολοκληρώνοντας τις οντότητες χαμηλού επιπέδου με τον οδηγό AAL, την οντότητα ICP και την οντότητα Διαχείρισης Συνδέσεων. Το λογισμικό εγκαταστάθηκε στο λειτουργικά πλήρως δημιουργώντας τα κατάλληλα tasks όπως αυτά θα τρέχουν σε κανονική λειτουργία (βλ. Παρ. 3.3.8). Για τον έλεγχο της σωστής λειτουργίας χρησιμοποιήθηκε ξανά η διάταξη της Εικόνας 4.3. Ο BLTG-A απέστειλε ένα cell EOP, αυτό διαβάστηκε από τον επεξεργαστή αποθηκεύτηκε στην ουρά ανάγνωσης και η PΧΔ ενημέρωσε το task που πραγματοποιεί τις λειτουργίες ICP και διαχείρισης. Αυτό το task ενεργοποιήθηκε και διάβασε το AAL πακέτο μέσω εντολής write του οδηγού AAL. Η σωστή λειτουργία του οδηγού εξακριβώθηκε συγκρίνοντας τα περιεχόμενα του πακέτου με αυτά του payload του cell. Καθώς το πακέτο που διαβάστηκε δεν ήταν ένα σωστό πακέτο ICP (τα cells του BLTG-A μεταφέρουν τυχαίες τιμές στο payload τους) χρειάστηκε να προστεθεί ένα ενδιάμεσο βήμα μεταξύ του οδηγού AAL και της λειτουργίας ICP κατά το οποίο το AAL πακέτο μετατράπηκε σε ICP πακέτο με παραμέτρους που καθορίστηκαν χειροκίνητα. Το νέο πακέτο διαβάστηκε από το ICP και εξακριβώθηκε η ορθή απόκριση του. Το πείραμα πραγματοποιήθηκε για όλους τους δυνατούς τύπους ICP πακέτων και ελέγχθηκαν όλες οι πιθανές αποκρίσεις του ICP. Έτσι εξακριβώθηκε η απάντηση σε ένα ICP\_PING πακέτο όπως και η απόκριση σε ένα πακέτο ICP διαχείρισης συνδέσεων. Στην δεύτερη περίπτωση επιβεβαιώθηκε η αποστολή των σωστών παραμέτρων προς την οντότητα διαχείρισης συνδέσεων. Τέλος

εξακριβώθηκε ότι η οντότητα διαχείρισης συνδέσεων πραγματοποίησε τις σωστές λειτουργίες σύμφωνα με τις προκαθορισμένες παραμέτρους που έλαβε από το ICP, τόσο για τις λειτουργίες εγκατάστασης όσο και κατάργησης μίας σύνδεσης.

2. Ελέγχθηκε η λειτουργία Ελέγχου Ροής ολοκληρώνοντας τις οντότητες χαμηλού επιπέδου και επιπέδου RA με το πρόγραμμα που εκτελεί τον αλγόριθμο. Το λογισμικό εγκαταστάθηκε στο λειτουργικά πλήρως δημιουργώντας τα κατάλληλα tasks όπως αυτά θα τρέχουν σε κανονική λειτουργία (βλ. Παρ. 3.3.8). Επίσης πραγματοποιήθηκαν με ειδικές συναρτήσεις (χειροκίνητα) οι αρχικοποιήσεις των πινάκων που χρησιμοποιεί ο αλγόριθμος ώστε να εγκατασταθούν στο σύστημα 6 ABR συνδέσεις με προκαθορισμένα VP/VC ενώ καθορίστηκε και μία συγκεκριμένη τιμή για το διαθέσιμο εύρος ζώνης. Για τον έλεγχο της σωστής λειτουργίας χρησιμοποιήθηκε ξανά η διάταξη της Εικόνας 4.3. Ο BLTG-A ρυθμίστηκε να αποστέλει κίνηση ABR με VP/VC ίδιο με τις εγκατεστημένες συνδέσεις και συγκεκριμένους ρυθμούς μετάδοσης προς την κάρτα και ο E.K. προγραμματίστηκε να εξάγει μόνο τα RM cells αυτών των συνδέσεων. Τα RM cells διαβάστηκαν από τον επεξεργαστή και ενεργοποιήθηκε σωστά το task που υλοποιεί την εξαγωγή των παραμέτρων και την εκτέλεση του αλγορίθμου. Η σωστή λειτουργία του αλγορίθμου εξακριβώθηκε ελέγχοντας (μέσω printf) τις τιμές εξόδου του αλγορίθμου και συγκρίνοντας τις με αυτές που θεωρητικά έχουν υπολογιστεί ανάλογα με τις παραμέτρους των συνδέσεων, τις παραμέτρους που μεταφέρει κάθε RM cell και την λογική ροή του αλγορίθμου. Τέλος οι τιμές εξόδου τοποθετήθηκαν με επιτυχία ξανά στα RM cells και αυτά εστάλειςαν προς την έξοδο της κάρτας. Η εξαγωγή των RM cells από την κάρτα επιβεβαιώθηκε μέσω του Analyzer του BLTG-A. Τέλος επιβεβαιώθηκε η παραπάνω σωστή διαδικασία σε κατάσταση συνεχούς λειτουργίας.
3. Σε τελικό στάδιο ελέγχθηκε η ολοκληρωμένη λειτουργία με ταυτόχρονη εκτέλεση των διαδικασιών ελέγχου ροής και διαχείρισης. Εξακριβώθηκε η ορθή παράλληλη λειτουργία καθώς και η εκτέλεση των tasks με βάση την προτεραιότητα που τους αποδόθηκε. Αυτό πραγματοποιήθηκε αποστέλοντας μέσω του BLTG-A κίνηση ABR για να ελεγχθεί ο έλεγχος ροής και ταυτόχρονα κίνηση CBR πολύ χαμηλού ρυθμού με EOP cells για να ελεγχθεί η παράλληλη λειτουργία των διαδικασιών διαχείρισης.

## 4.4 Μετρήσεις-Αποτελέσματα

### 4.4.1 Λειτουργία Διαχείρισης

Η ταχύτητα εκτέλεσης της λειτουργίας διαχείρισης μετρήθηκε ως εξής:

Χρησιμοποιήθηκε ο BLTG-A με τον οποίο παραγόταν cells EOP με διάφορους ρυθμούς συνεχώς αυξανόμενους. Με αυτόν τον τρόπο παρατηρήθηκε ο μέγιστος ρυθμός με τον οποίο οι λειτουργίες διαχείρισης μπορούν να εξυπηρετούν

εισερχόμενα cells πρώτου γεμίσουν οι ουρές και αρχίσουν να χάνονται cells. Αυτό πραγματοποιήθηκε συγκρίνοντας για διάφορους ρυθμούς αποστολής, τον αριθμό των cells-πακέτων που αποστέλονται από τον BLTG-A σε μεγάλο χρονικό διάστημα με τον αριθμό των cells-πακέτων που εξυπηρετήθηκαν από την οντότητα διαχείρισης στο ίδιο διάστημα. Αυτός ο μέγιστος ρυθμός μετρήθηκε:

$$\text{Ρυθμός Επεξεργασίας cells Διαχείρισης} = 1100 \text{ cells/sec}$$

#### 4.4.2 Λειτουργία Ελέγχου Ροής

Για την μέτρηση της απόδοσης των λειτουργιών διαχείρισης χρησιμοποιήθηκε ο BLTG-A με τον οποίο παραγόταν ABR κίνηση με διάφορους ρυθμούς συνεχώς αυξανόμενους. Στην παραγόμενη κίνηση ρυθμίστηκε ο ρυθμός αποστολής RM cells καθορίζοντας τον αριθμό των data cells που αποστέλονται μεταξύ δύο RM cells. Με αυτόν τον τρόπο παρατηρήθηκε ο μέγιστος ρυθμός με τον οποίο οι λειτουργίες ελέγχου ροής μπορούν να εξυπηρετούν εισερχόμενα RM cells πρώτου γεμίσουν οι ουρές και αρχίσουν να χάνονται cells. Αυτό πραγματοποιήθηκε συγκρίνοντας για διάφορους ρυθμούς αποστολής, τον αριθμό των RM cells πακέτων που αποστέλονται από τον BLTG-A σε μεγάλο χρονικό διάστημα με τον αριθμό των RM cells που εξυπηρετήθηκαν από την οντότητα ελέγχου ροής στο ίδιο διάστημα.

Με τον παραπάνω τρόπο μετρήθηκαν διαδοχικά οι διάφοροι ρυθμοί εξυπηρέτησης, εκτελώντας συγκεκριμένα μέρη της επεξεργασίας των RM cells. Αυτό είχε ως σκοπό να μετρηθεί η συμβολή κάθε βασικού μέρους της επεξεργασία και να προσδιοριστούν με αυτόν τον τρόπο τα bottlenecks στην επεξεργασία των RM cells ώστε να προταθούν δυνατότητες βελτίωσης της απόδοσης. Οι περιπτώσεις που εξετάστηκαν είναι οι εξής:

1. μέτρηση του ρυθμού επεξεργασίας σε ολοκληρωμένο σύστημα το οποίο διαβάσει cells, τα επεξεργάζεται, τα τοποθετεί σε ουρά εξόδου και τέλος τα γράφει ξανά στο φυσικό στοιχείο (E.K.)
2. μέτρηση του ρυθμού επεξεργασίας χωρίς τις διαδικασίες τοποθέτησης σε ουρά εξόδου
3. μέτρηση του μέγιστου ρυθμού εγγραφής cells στον E.K.
4. μέτρηση του μέγιστου ρυθμού ανάγνωσης cells από τον E.K.
5. μέτρηση του μέγιστου ρυθμού ανάγνωσης και επεξεργασίας χωρίς την εγγραφή cells στον E.K.

Στον Πίνακα 6 φαίνονται οι μετρήσεις που έγιναν για όλες τις περιπτώσεις. Στην πρώτη στήλη περιγράφεται το κομμάτι του λογισμικού επεξεργασίας RM cells που μετρήθηκε, στη δεύτερη στήλη βρίσκεται ο ρυθμός εξυπηρέτησης RM cells, στην τρίτη στήλη φαίνεται ο μέσος χρόνος εξυπηρέτησης ενός RM cell για κάθε περίπτωση ο οποίος προκύπτει από τον ρυθμό εξυπηρέτησης.

<i>Περίπτωση</i>	<i>Ρυθμός Εξυπηρέτησης(cells/s)</i>	<i>Μέσος χρόνος εξυπηρέτησης(μs)</i>
1	1460	685
2	2800	357
3	7700	130
4	7400	135
5	4500	222

Πίνακας 6 Μετρήσεις

Τα πρότυπα ελέγχου ροής του ATM forum καθορίζουν ότι ο ελάχιστος ρυθμός αποστολής RM cells από μία πηγή είναι **10 cells/s**. Άρα αν το λογισμικό ελέγχου ροής μπορεί να εξυπηρετήσει A cells το δευτερόλεπτο τότε ο μέγιστος αριθμός πηγών που μπορούν να εξυπηρετηθούν από το σύστημα είναι  $A/10$ .

Με βάση αυτήν την παρατήρηση, από τον ??? προκύπτουν τα εξής αποτελέσματα:

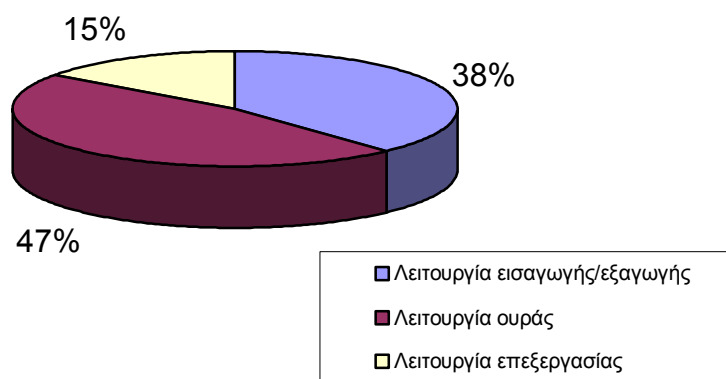
- Η λειτουργία ελέγχου ροής όπως σχεδιάστηκε και υλοποιήθηκε σε αυτή την εργασία μπορεί να εξυπηρετήσει το πολύ μέχρι 146 ABR συνδέσεις (περίπτωση 1).
- Εάν αφαιρεθεί η ουρά εξόδου και οι λειτουργίες που σχετίζονται με αυτή τότε το σύστημα μπορεί να εξυπηρετήσει μέχρι 280 ABR συνδέσεις (περίπτωση 1). Η αύξηση είναι της τάξης του 92%.
- Ο μέσος χρόνος που απαιτούν οι λειτουργίες ουράς είναι 328μs (από περίπτωση 1,2)
- Ο μέσος χρόνος που απαιτούν οι λειτουργίες εισαγωγής/εξαγωγής cells και επικοινωνίας με το φυσικό στοιχείο είναι 265μs (από περιπτώσεις 1,2 )
- Ο χρόνος που απαιτεί μόνη της η λειτουργία εκτέλεσης του αλγορίθμου μαζί με τις λειτουργίες του επιπέδου RA είναι περίπου 90 μs. (από περιπτώσεις 3,5 και 1,2,3,4). Αυτό σημαίνει ότι η επεξεργασία των RM cells μόνη της μπορεί να εξυπηρετεί 11000 c/s και συνεπώς 1100 συνδέσεις.

Στον Πίνακα 7 αναγράφεται ο αριθμός των συνδέσεων που μπορούν να εξυπηρετηθούν ανάλογα με τις λειτουργίες που εκτελεί το λογισμικό

<i>Λειτουργία</i>	<i>Αριθμός Εξυπηρετίσιμων Συνδέσεων</i>
Ολοκληρωμένη Λειτουργία	146
Λειτουργία χωρίς ουρά	280
Λειτουργία χωρίς εισγωγή/εξαγωγή cells	1100

**Πίνακας 7 Συμμετοχή λειτουργιών στον χρόνο εξυπηρέτησης**

Στην Εικόνα 45 φαίνεται η συμμετοχή των διαφόρων λειτουργιών του λογισμικού στον συνολικό χρόνο εξυπηρέτησης και επεξεργασίας ενός RM cell κατά την διάρκεια της ολοκληρωμένης λειτουργίας.



**Εικόνα 45 Ποσοστιαία συμμετοχή των λειτουργιών της εξυπηρέτησης RM cells**

## 5 Συμπεράσματα – Προτάσεις - Επισκόπηση

### 5.1 Συμπεράσματα

Κατά την διάρκεια της παρούσας εργασίας σχεδιάστηκαν και υλοποιήθηκαν όλες οι οντότητες που είναι απαραίτητες για την λειτουργία και διαχείριση της κάρτας Εξυπηρετητή ABR, καθώς και λογισμικό για την εκτέλεση πολύπλοκου αλγορίθμου ελέγχου ροής. Το λογισμικό αυτό σχεδιάστηκε και υλοποιήθηκε με τρόπο που να είναι αποδοτικός εύκολα προσαρμόσιμος και επεκτάσιμος σε μελλοντικές αναβαθμίσεις του συστήματος.

Υλοποιήθηκε εγκαταστάθηκε και ελέγχθηκε η λειτουργία επικοινωνίας της κάρτας με το υπόλοιπο σύστημα καθώς και οι λειτουργίες διαχείρισης. Αυτές οι λειτουργίες δοκιμάστηκαν στο πραγματικό υλικό και επιβεβαιώθηκε η ορθή λειτουργία τους σε όλες τις πιθανές περιπτώσεις. Επιπλέον μετρήθηκε η απόδοση αυτών των λειτουργιών (παρ. 4.4.1). Με βάση αυτές τις μετρήσεις βγαίνει το συμπέρασμα ότι ο χρόνος εκτέλεσης των λειτουργιών διαχείρισης και επικοινωνίας που μετρήθηκε της



τάξης του 1ms επαρκεί για την αποδοτική λειτουργία του συστήματος καθώς η αναμενόμενη συχνότητα εκτέλεσης της λειτουργίας διαχείρισης είναι της τάξης των δευτερολέπτων ή δεκάδων δευτερολέπτων.

Σε δεύτερο επίπεδο σχεδιάστηκε, υλοποιήθηκε, εγκαταστάθηκε και ελέγχθηκε η πολύπλοκη και απαιτητική λειτουργία ελέγχου ροής. Η συγκεκριμένη λειτουργία δοκιμάστηκε στο πραγματικό υλικό και επιβεβαιώθηκε ο ορθός χειρισμός των RM cells όσο και σωστή εκτέλεση του αλγορίθμου ελέγχου ροής. Επιπλέον μετρήθηκε η απόδοση του συστήματος και οι δυνατότητες εξυπηρέτησης που παρέχει με την υλοποίηση που περιγράφηκε στην παρούσα εργασία (κεφ 4.4.2). Η απόδοση αυτή προέκυψε επαρκής για την αποδοτική λειτουργία του συστήματος (η κάρτα Εξυπηρετητή ABR στην αρχική της μορφή μερικές εκατοντάδες συνδέσεων). Ωστόσο δεν είναι ικανοποιητική για ένα επεκτάσιμο σύστημα το οποίο μπορεί να ικανοποιεί μέχρι και 32000 συνδέσεις γεγονός που οφείλεται στους περιορισμένους πόρους του που δεν επερακούν για εκτέλεση λειτουργιών πραγματικού χρόνου μεγάλων ταχυτήτων εξολοκήρου σε λογισμικό. Όμως όπως προκύπτει από τις μετρήσεις το bottleneck στην εκτέλεση του ελέγχου ροής βρίσκεται στις λειτουργίες εξαγωγής/εισαγωγής cells μέσω του Επεξεργαστή Κελιών καθώς και αποθήκευσής τους στην μνήμη του επεξεργαστή. Και αυτό γιατί ο χρόνος που απαιτείται για αυτές τις λειτουργίες είναι κατά πολύ μεγαλύτερος από τον χρόνο επεξεργασίας των cells και επίσης είναι τόσος που να μην αρκεί όταν η συχνότητα εκτέλεσης αυτών των λειτουργιών είναι σχετικά μεγάλη (μεγαλύτερη από 2000 φορές το δευτερόλεπτο).

Από τα παραπάνω, βγαίνει το συμπέρασμα ότι για να βελτιωθεί η αποδοτικότητα της διαδικασίας ελέγχου ροής και να γίνει περισσότερο επεκτάσιμη θα πρέπει να μεταφερθούν οι λειτουργίες του επιπέδου ATM από το λογισμικό στο υλικό. Αυτό μπορεί να πραγματοποιηθεί αν στην Μ.Ε.Κ.Α. προστεθεί η δυνατότητα να επεξεργάζεται RM cells να εξάγει τις παραμέτρους που απαιτούνται και να μεταφέρει μόνο αυτές (μεταφορά 6 bytes) στον επεξεργαστή και στο λογισμικό αντί να μεταφέρονται ολόκληρα cells (60 bytes για το format του Ε.Κ.). Επίσης στην αντίθετη κατεύθυνση θα απαιτείται να μεταφέρεται από τον επεξεργαστή προς το υλικό μόνον η έξοδος του αλγορίθμου (2 bytes) αντί να μεταφέρεται ολόκληρο το ανανεωμένο cell (60 bytes).

Πολύ σημαντικό συμπέρασμα σε αυτό το σημείο είναι ότι η υλοποίηση του λογισμικού για τον έλεγχο ροής όπως περιγράφηκε στα προηγούμενα κεφάλαια είναι πολύ εύκολα προσαρμόσιμη στην ενδεχόμενη σχεδίαση που περιγράφηκε προηγουμένως χάρη στην αρθρωτή (modular) δομή του. Η προσαρμογή μπορεί να πραγματοποιηθεί πολύ σύντομα αφαιρώντας από την διαδικασία ελέγχου ροής όλες τις λειτουργίες χαμηλού επιπέδου και επιπέδου ATM και μετατρέποντας τις λειτουργίες του επιπέδου RA έτσι ώστε να αλληλεπιδρούν απευθείας με την διεπαφή επεξεργαστή της Μ.Ε.Κ.Α. Μία γενική εκτίμηση είναι ότι αυτό μπορεί να πραγματοποιηθεί με αλλαγή μόνο μερικών δεκάδων γραμμών από τον συνολικό υλοποιημένο και ελεγμένο κώδικα.

## 5.2 Επισκόπηση εργασιών

Η διάρκεια της παρούσας μεταπτυχιακής εργασίας ήταν 18 μήνες. Για τους σκοπούς της πραγματοποιήθηκαν μέσα σε αυτό το διάστημα 13 γενικές κατηγορίες δουλειάς. Αυτές οι κατηγορίες καθώς και η ποσοστιαία συμμετοχή τους στον συνολικό φόρτο της μεταπτυχιακής εργασίας αναφέρονται στη συνέχεια:

1. *Μελέτη και κατανόηση της αρχιτεκτονικής του Κόμβου Πρόσβασης και της αρχιτεκτονικής της κάρτας Εξυπηρετητή ABR.*  
Συμμετοχή 6%
2. *Σχεδίαση του PCB της κάρτας Εξυπηρετητή Κίνησης ABR*  
Συμμετοχή 10%
3. *Μελέτη λειτουργίας και διεπαφών των φυσικών στοιχείων που υλοποιούν την κάρτα*  
Συμμετοχή 8%
4. *Εκμάθηση λειτουργικού συστήματος (VxWorks) του επεξεργαστή της κάρτας*  
Συμμετοχή 6%
5. *Καθορισμός και εγκατάσταση εργαλείων για την ανάπτυξη του λογισμικού.*  
Συμμετοχή 3%
6. *Μελέτη, κατανόηση και μετατροπή του υπάρχοντος κώδικα.*  
Συμμετοχή 3%
7. *Σχεδίαση Λογισμικού*  
Συμμετοχή 5%
8. *Υλοποίηση λογισμικού, συγγραφή κώδικα*  
Συμμετοχή 8%
9. *Εγκατάσταση διατάξεων για τον έλεγχο της κάρτας*  
Συμμετοχή 3%
10. *Έλεγχος ορθής λειτουργίας της κάρτας και των διεπαφών των φυσικών στοιχείων με τον επεξεργαστή*  
Συμμετοχή 24%
11. *Εγκατάσταση διατάξεων για τον έλεγχο του λογισμικού*  
Συμμετοχή 3%
12. *Εγκατάσταση λειτουργικού συστήματος και λογισμικού στην κάρτα.*  
Συμμετοχή 3%
13. *Έλεγχος λογισμικού και εξαγωγή συμπερασμάτων*  
Συμμετοχή 18%

### **5.3 Προτάσεις για μελλοντική εργασία**

Οι μελλοντικές εργασίες που μπορούν ή χρειάζεται να πραγματοποιηθούν βραχυπρόθεσμα ή μακροπρόθεσμα και αφορούν την κάρτα Εξυπηρετητή ABR το λογισμικό της αναφέρονται στη συνέχεια:

1. Εγκατάσταση της κάρτας στο συνολικό σύστημα του μεταγωγέα για να ελεγχθούν οι διαδικασίες επικοινωνίας και διαχείρισης σε ολοκληρωμένες πραγματικές συνθήκες
2. Έλεγχος του λογισμικού διαχείρισης όταν αυτό εκτελείται πάνω από διαφορετικό φυσικό στοιχείο.
3. Επέκταση του πρωτοκόλλου επικοινωνίας ώστε να υποστηρίζει περισσότερες εντολές (π.χ. εντολές ανάγνωσης μνήμης).
4. Υλοποίηση εναλλακτικών αλγορίθμων ελέγχου ροής
5. Υλοποίηση του απαραίτητου υλικού στην Μ.Ε.Κ.Α., εξέλιξη της διεπαφής της με τον επεξεργαστή και μετατροπή του λογισμικού ώστε τα χρονοβόρα μέρη της λειτουργίας του ελέγχου ροής να μεταφερθούν στο υλικό με σκοπό να βελτιωθεί η απόδοση της.
6. Υλοποίηση και εγκατάσταση στην κάρτα λογισμικού για την εκτέλεση επιπλέον λειτουργιών που θα συμβάλουν στην αυτόνομη λειτουργία της ως κάρτα δικτύου (π.χ. ILMI, signaling)

## Παραρτήματα

## ***Παράρτημα Α - Η Μονάδα Εξυπηρέτησης Κίνησης ABR (M.E.K.A.)***

### ***Εισαγωγή***

Το σύστημα που θα υλοποιήσουμε σε αυτό το έργο θα εξυπηρετεί σχετικά μικρό αριθμό χρηστών (το πολύ λίγες δεκάδες). Σε αυτή την περίπτωση η υποστήριξη ακόμα και λίγων εκατοντάδων ουρών θα ήταν αρκετή για την υποστήριξη πολλαπλών συνδέσεων με κίνηση ABR ανά χρήστη. Η μονάδα εξυπηρέτησης κίνησης ABR θα υποστηρίζει λίγες χιλιάδες ουρές, έτσι ώστε να μπορεί να χρησιμοποιηθεί χωρίς αλλαγές και σε μεγαλύτερα συστήματα.

Από πλευράς χώρου για την αποθήκευση των δεδομένων, δεδομένης της χρήσης συνδέσεων με ταχύτητα 155 Mbps, η χρήση μνήμης μεγέθους λίγων δεκάδων Mbytes αρκεί για την αποθήκευση του συνόλου της κίνησης για μερικά δευτερόλεπτα. Δεδομένου ότι η μνήμη θα εξυπηρετεί μόνο κίνηση τύπου ABR ακόμα και λίγα Mbytes θα είναι ικανά να ανταποκριθούν σε σημαντικές διακυμάνσεις της διαθεσιμότητας του bandwidth. Η δε χρήση μνημών τύπου SDRAM επιβάλλει ένα ελάχιστο μέγεθος 32Mbytes, το οποίο θα είναι υπεραρκετό για τις ανάγκες της κίνησης ABR.

### **Ο Προγραμματιστής Cells (Cell Scheduler)**

Η Μονάδα Cell Scheduler του ABRSU δέχεται cells από τις συσκευές CubitPro μέσω μιας διασύνδεσης UTOPIA, αντικαθιστά την Επικεφαλίδα Δρομολόγησης του CellBus (CellBus Routing Header), που τοποθετείται μπροστά από το cell από το CubitPro της Κάρτας που έλαβε αρχικά το πακέτο, με την Επικεφαλίδα Δρομολόγησης Tandem (Tandem Routing Header) που παρέχεται από το ίδιο CubitPro και περιέχει πληροφορίες για τον τελικό προορισμό του Cell. Στη συνέχεια μεταφράζει το VP/VC του cell σε QID (Queue ID) και στέλνει το πακέτο μαζί με το QID στο Διαχειριστή Ουρών (Queue Manager) για αποθήκευση. Μια μονάδα μνήμης SRAM χρησιμοποιείται εδώ για την αποθήκευση των πινάκων μετάφρασης από VP/VC σε QID.

Σημείωση: Στην περίπτωση που τα bits του Tandem Routing Header δεν είναι όλα απαραίτητα, μπορούν να χρησιμοποιηθούν για να σταλεί το QID από την Κάρτα Γραμμής που έλαβε το cell. Έτσι η μετάφραση από VP/VC σε QID γίνεται από την κάρτα γραμμής απλοποιώντας την δουλειά του Cell Scheduler.

Στην άλλη κατεύθυνση, ο Cell Scheduler ζητά και λαμβάνει από τον Queue Manager, cells που ανήκουν σε δεδομένες ουρές, με βάση κάποια πολιτική δρομολόγησης (π.χ. Round Robin) και στη συνέχεια τα προωθεί στις συσκευές CubitPro.

### **Ο Διαχειριστής Ουρών (Queue Manager)**

Ο Διαχειριστής Ουρών, τμήμα του ABR SU, δέχεται cells από το Cell Scheduler μαζί με το QID που καθορίζει την αντίστοιχη ουρά που κάθε cell θα αποθηκευτεί με πολιτική FIFO. Μια μονάδα μνήμης SRAM χρησιμοποιείται εδώ για αποθήκευση πληροφοριών διαχείρισης των ουρών (head και tail pointers), ενώ μια μνήμη SDRAM χρησιμοποιείται για την αποθήκευση των ίδιων των Cell.

Ο Queue Manager απαντά στις αιτήσεις του Cell Scheduler βγάζει από τις ουρές cells για μετάδοση. Ακόμη απαντά στις αιτήσεις του επεξεργαστή MPC860 για την δημιουργία νέων ουρών VP/VC (CAC – Call Admission Control για μια νέα σύνδεση ABR ) με την δέσμευση ενός από το πεπερασμένο πλήθος ελεύθερων QIDs.

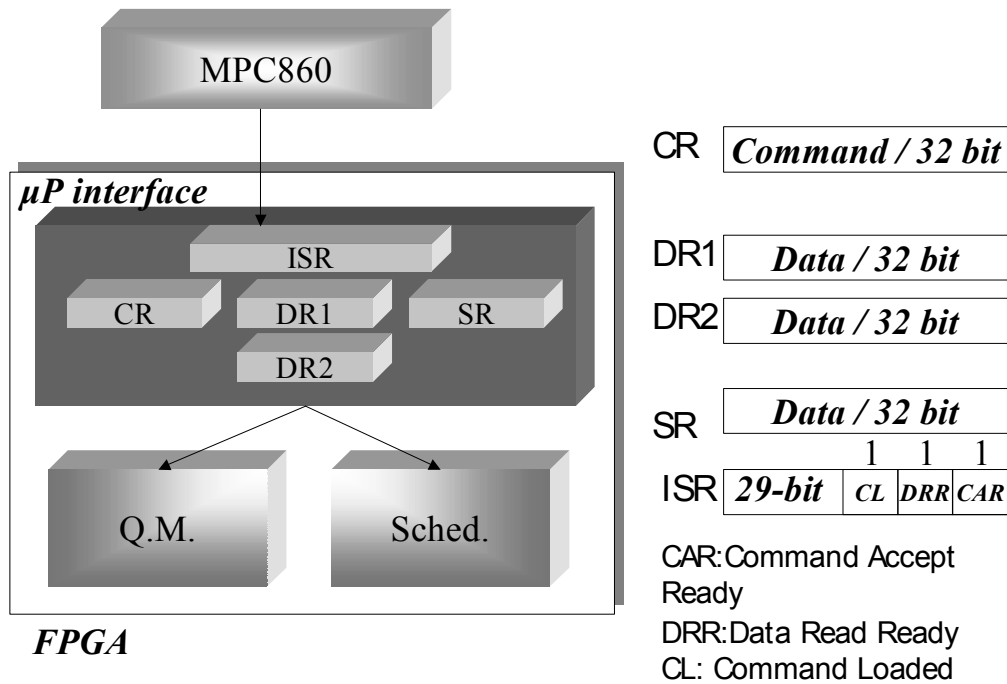
Τέλος ο διαχειριστής ουρών διαχειρίζεται και τον ελεύθερο χώρο στην μνήμη αποθήκευσης δεδομένων μέσω μίας ουράς ελεύθερου χώρου (free-list) και ενημερώνει την δομή αυτή κάθε φορά που ένα cell αποθηκεύεται ή μεταδίδεται.

### **Διεπαφή M.E.K.A. με τον εσωτερικό επεξεργαστή ελέγχου.**

#### ***Καταχωρητές για την διασύνδεση με τον επεξεργαστή***

Κάθε φορά που ο επεξεργαστής θέλει να επικοινωνήσει με την FPGA παράγει μια 32-bit λέξη που περιέχει το opcode της εντολής καθώς και τις παραμέτρους της. Αυτή η λέξη τοποθετείται σε κατάλληλους καταχωρητές εσωτερικά της FPGA.

Οι καταχωρητές που χρησιμοποιούνται για την επικοινωνία μικροεπεξεργαστή-FPGA φαίνονται στο **Error! Reference source not found.** Στο Σχήμα 2 φαίνεται η διεύθυνση που αντιστοιχεί στο εσωτερικό memory map για τον κάθε καταχωρητή.



Σχήμα 1 Διεπαφή επεξεργαστή της Μ.Ε.Κ.Α.

<i>Όνομα Καταχωρητή</i>	<i>Συντομογραφία</i>	<i>Εύρος</i>	<i>Εσωτερ.Διεύθυνση</i>
Synchronisation/Status Register	SSR	32 bits	0x00
Command Register	CR	32 bits	0x04
Data Register 1	DR1	32 bits	0x08
Data Register 2	DR2	32 bits	0x0c
Scheduling Register	SR	32 bits	0x10

Σχήμα 2 Το εσωτερικό memory map της FPGA

*Command Register:* Αυτός ο καταχωρητής δέχεται την εντολή προς τον επεξεργαστή.

*Data Register 1:* Αυτός ο καταχωρητής κρατάει 32 περισσότερο σημαντικά ψηφία των δεδομένων που πρέπει να γραφούν στην εξωτερική μνήμη σε περίπτωση εντολής Write, ή των αποτελέσμων εντολών για όλες τις υπόλοιπες εντολές.

*Data Register 2:* Αυτός ο καταχωρητής κρατάει 32 λιγότερο σημαντικά ψηφία των δεδομένων που πρέπει να γραφούν στην εξωτερική μνήμη σε περίπτωση εντολής Write, ή των αποτελέσμων εντολών για όλες τις υπόλοιπες εντολές.

*Synchronisation/Status Register*: Αυτός ο καταχωρητής δείχνει την κατάσταση της διεπαφής μεταξύ μικροεπεξεργαστή και FPGA (command ready, data ready).  
*Scheduling Register*: Σε αυτόν τον καταχωρητή γράφεται η πληροφορία για τα διαστήματα εξυπηρέτησης (service intervals) των διαφόρων ροών (flows) που δίδεται στον scheduler.

### Αλληλεπίδραση Επεξεργαστή-FPGA

Η αλληλεπίδραση του μικροεπεξεργαστή με την FPGA γίνεται αποκλειστικά μέσω εντολών. Ο μικροεπεξεργαστής παράγει εντολές για την FPGA και τις παρέχει σε αυτήν μέσω του καταχωρητή CR. Οι εντολές αυτές έχουν συγκεκριμένο format και η τοποθέτηση τους μέσα στον καταχωρητή CR γίνεται με προκαθορισμένο τρόπο. Οι εντολές, το format, και η τοποθέτηση στον καταχωρητή εντολών παρουσιάζονται στη συνέχεια:

- **OpenConnection**: Η εντολή παρέχεται στην FPGA για να την ενημερώσει ότι έχει εγκατασταθεί μία νέα ABR σύνδεση στον μεταγωγέα.

4 bits	8 bits	16 bits
Opcode	Flow Group Id	Flow Id

Σχήμα 3 Format της εντολής OpenConnection

- *Opcode*: Ο κωδικός της εντολής (0x01 για OpenConnection)
- *Flow Group ID*: Η ομάδα ροών στην οποία ανήκει η νέα σύνδεση. Η ομάδα ροών είναι η οντότητα σύμφωνα με την οποία καθορίζεται το scheduling της κυκλοφορίας εξόδου. Συχνά η ομάδα ροών ταυτίζεται με το φυσική γραμμή εξόδου (Output Physical Link) η οποία περιέχει όλες τις ροές (connections) της ομάδας.
- *Flow ID*: Είναι ο αριθμός που προσδιορίζει την σύνδεση. Αυτός ο αριθμός είναι ένα εσωτερικό προσδιοριστικό (Internal Identifier) του μεταγωγέα το οποίο προκύπτει από το ζευγάρι VP/VC της σύνδεσης. Ως Flow ID χρησιμοποιείται το προσδιοριστικό ICI το οποίο τοποθετείται σε κάθε cell μέσω της VP/VC μετάφρασης που πραγματοποιείται στην κάρτα εισόδου (κάρτα ATM)

### Command Register (CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Opcode(0x01)				0				Flow Group ID							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Flow ID															

Σχήμα 4 Περιεχόμενο καταχωρητή CR για την εντολή OpenConnection



- **CloseConnection:** Η εντολή παρέχεται στην FPGA για να την ενημερώσει ότι μία υπάρχουσα ABR σύνδεση έχει κλείσει.

4 bits	16 bits
Opcode	Flow ID

**Σχήμα 5 Format εντολής CloseConnection**

- ο Οι παράμετροι της εντολής περιγράφονται στην εντολή OpenConnection. Το opcode είναι 0x02.

Command Register (CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Opcode(0x02)				0				0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Flow ID															

**Σχήμα 6 Περιεχόμενο καταχωρητή CR για την εντολή CloseConnection**

- **ReadCnt:** Η εντολή παρέχεται στην FPGA ως αίτηση ανάγνωσης των εσωτερικών μετρητών της. Αυτοί οι μετρητές κρατούν το μήκος της ουράς της κάθε σύνδεσης. Η εντολή ζητάει την τιμή του μετρητή που αντιστοιχεί στην σύνδεση που περιγράφεται από την παράμετρο FlowID της εντολής. Η FPGA αποκρίνεται στην συγκεκριμένη εντολή επιστρέφοντας την τιμή του μετρητή (και συνεπώς το μήκος της ουράς) μέσω των καταχωρητών DR\*.

4 bits	16 bits
Opcode	Flow ID

**Σχήμα 7 Format εντολής ReadCnt**

- ο Το opcode της εντολής είναι 0x03, η παράμετρος FlowID περιγράφηκε στις προηγούμενες εντολές.

**Command Register (CR)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Opcode(0x03)				0				0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Flow ID															

**Σχήμα 8 Τιμή καταχωρητή CR για την εντολή ReadCnt**

- **SetWatermarks:** Με αυτήν την εντολή ενημερώνεται η FPGA για τα επίπεδα ασφαλείας που αφορούν τα μήκη των ουρών. Αυτή η πληροφορία είναι απαραίτητη για το κομμάτι που εκτελεί τις λειτουργίες flow control για τις διάφορες συνδέσεις. Αυτό το κομμάτι εκτελεί την λειτουργία relative rate marking των RM cells η οποία για να εκτελεστεί χρειάζεται να καθοριστούν

τα όρια του μήκους της ουράς που αντιστοιχεί στην κάθε σύνδεση πάνω από τα οποία θα πραγματοποιείται η λειτουργία marking. Η εντολή SetWmarks ακολουθεί σχεδόν πάντα μία εντολή OpenConnection για να αναθέσει επιτρεπόμενα μήκη ουρών σε μία νέα σύνδεση. Μπορεί όμως να δοθεί μία τέτοια εντολή μεμονωμένα σε περίπτωση που τα όρια της ουράς μίας σύνδεσης πρέπει για κάποιον λόγο να αλλάξουν.

4 bits	8bits	4bits	16 bits
Opcode	HWm	WmOff	Flow ID

**Σχήμα 9 Format εντολής SetWmarks**

- Το opcode της εντολής είναι 0x04
- HWm (High Watermark): Για την διαδικασία marking των RM cells απαιτούνται δύο όρια μήκους ουρών. Η παράμετρος HWm είναι το υψηλότερο από τα δύο όρια και εκφράζεται σε αριθμό cells???.
- WmOff(Watermark Offset): Είναι η τιμή που καθορίζει το χαμηλό όριο. Εκφράζεται ως η απόλυτη διαφορά του υψηλού ορίου από το χαμηλό.
- FlowID: η παράμετρος που περιγράφηκε στις προηγούμενες εντολές.

**Command Register (CR)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Opcode(0x04)				HWm								WmOff			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FlowId															

**Σχήμα 10 Τιμή του καταχωρητή CR στην περίπτωση εντολής SetWmarks**

- Read: Είναι η γενική εντολή ανάγνωσης μίας των περιεχομένων μίας διεύθυνσης της μνήμης που χρησιμοποιεί η FPGA. Η FPGA αποκρίνεται στην συγκεκριμένη εντολή επιστρέφοντας το περιεχόμενο της διεύθυνσης που δίδεται μέσω των καταχωρητών DR\*.

4 bits	16 bits
Opcode	Address

**Σχήμα 11 Το format της εντολής Read**

- Το opcode της εντολής είναι 0x05.
- Address: Η 16-bit διεύθυνση της μνήμης της FPGA η οποία θα διαβαστεί.

### Command Register (CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Opcode(0x05)				0				0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Address															

Σχήμα 12 Η τιμή του καταχωρητή CR σε περίπτωση εντολής Read

- Write: Είναι η γενική εντολή εγγραφής σε μία διεύθυνση της μνήμης που χρησιμοποιεί η FPGA.

4 bits	16 bits	64 bits
Opcode	Address	Data

Σχήμα 13 Το format της εντολής Write

- Opcode: 0x06
- Address: Η 16-bit διεύθυνση της μνήμης της FPGA στην οποία θα γίνει η εγγραφή
- Data : Τα 64-bit δεδομένα που θα γραφούν στην καθορισμένη διεύθυνση

Η εντολή write χρησιμοποιεί τον καταχωρητή CR προκειμένου να παρέχει την εντολή (opcode) και την διεύθυνση στην οποία θα γίνει η εγγραφή και τους καταχωρητές DR1,2 για να παρέχει τα δεδομένα (μήκους 64 bits) που θα γραφούν στην συγκεκριμένη διεύθυνση.

### Command Register (CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Opcode(0x06)				0				0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Address															

Σχήμα 14 Η τιμή του καταχωρητή CR σε περίπτωση εντολής Write

#### 5.3.1.1 Data Register (DR1,2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data															

Σχήμα 15 Η τιμή των καταχωρητών DR1,2 σε περίπτωση εντολής Write

- **SetLinkBW** (για τον scheduler): Η εντολή αυτή ενημερώνει τον Scheduler εσωτερικά της FPGA για το διάστημα εξυπηρέτησης (service interval) το οποίο αντιστοιχεί σε ένα Σύνολο Ροών (Flow Group). Το διάστημα εξυπηρέτησης προκύπτει από αριθμητικό μετασχηματισμό ποσοτήτων που αφορούν τους διαθέσιμους πόρους του Συνόλου Ροών. Η ενημέρωση την FPGA μέσω αυτής της εντολής παραγματοποιείται οπωσδήποτε κάθε φορά που εγκαθίσταται η παύει μία νέα σύνδεση οποιουδήποτε τύπου (CBR,VBR,ABR). Συνεπώς σε περίπτωση σύνδεσης ABR η εντολή SetLinkBW ακολουθεί την εντολή OpenConnection (ή CloseConnection), ενώ σε περίπτωση σύνδεσης άλλου τύπου η εντολή εκτελείται μόνη της.

Αντίθετα με τις υπόλοιπες εντολές, η SetLinkBW παρέχεται στην FPGA μέσω του καταχωρητή Scheduling Register (SR) και όχι μέσω του καταχωρητή Command Register (CR).

### **Λειτουργία διεπαφής μικροεπεξεργαστή-FPGA**

Ο μικροεπεξεργαστής επικοινωνεί με την FPGA μέσω εντολοών που δημιουργεί και γράφει σε αυτή και μέσω δεδομένων που διαβάζει από αυτή. Επειδή προβλέπεται ο φόρτος της FPGA να είναι μεγάλος, η απόκριση της στις εντολές του μικροεπεξεργαστή, σίγουρα δεν θα είναι άμμεση. Γι αυτόν τον λόγο η διοχέτευση εντολών προς τον επεξεργαστή δεν θα πρέπει να είναι αυθαίρετη. Ο επεξεργαστής θα πρέπει να παρέχει μια νέα εντολή στην FPGA αφότου αυτή επεξεργαστεί την προηγούμενη. Επίσης η παραγωγή και ανάγνωση αποτελεσμάτων που παρέχει η FPGA είναι ασύγχρονη ως προς την λειτουργία τόσο της FPGA όσο και του επεξεργαστή. Για να εξασφαλιστεί ο απαραίτητος συγχρονισμός στις λειτουργίες παροχής εντολών και ανάγνωσης αποτελεσμάτων, θα πρέπει να υπάρχει η κατάλληλη πληροφορία συγχρονισμού μεταξύ επεξεργαστή και FPGA. Αυτή η πληροφορία τοποθετείται στον καταχωρητή SSR ο οποίος είναι καταχωρητής ελέγχου της διεπαφής επεξεργαστή-FPGA και στον οποίο έχουν πλήρη πρόσβαση (ανάγνωση και εγγραφή) και τα δύο μέρη της διεπαφής..

### **Συγχρονισμός μικροεπεξεργαστή και FPGA**

**Synchronisation Status Register (SSR)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	SAR	0	CL	0	DRR	0	CAR

**Σχήμα 16 Δομή του Synchronization Status Register (SRR)**

Η δομή του καταχωρητή SRR φαίνεται στο Σχήμα 16. Η χρήση αυτού του καταχωρητή και η γενικότερη λειτουργία του συγχρονισμού έχει ως εξής:

Όταν ο επεξεργαστής έχει έτοιμη μία εντολή να δώσει στην FPGA κοιτάει το Command Accept Ready (CAR) bit του ISR. Αν αυτό έχει τεθεί από την FPGA τότε

αυτή έχει ή επεξεργαστεί την προηγούμενη εντολή οπότε το περιεχόμενο του καταχωρητή CR είναι δυνατόν να πάρει νέα τιμή (νέα εντολή), είναι συνεπώς η FPGA έτοιμη να δεχθεί νέα εντολή οπότε ο επεξεργαστής γράφει την εντολή στον κατάλληλο καταχωρητή (CR) και μηδενίζει το CAR bit, ενώ ταυτόχρονα θέτει το Command Loaded (CL) bit για να ενημερώσει την FPGA ότι μία νέα εντολή φορτώθηκε και ότι ο καταχωρητής CR περιέχει έγκυρη νέα εντολή. Η FPGA την επόμενη φορά που θα είναι σε θέση να απορροφήσει την νέα εντολή κοιτάζει το CL bit του SSR και αν αυτό έχει τεθεί τότε την έγκυρη εντολή που βρίσκεται στον CR την διαβάζει και την εκτελεί. Αφού την διαβάσει τότε θέτει ξανά το bit CAR για να ενημερώσει τον επεξεργαστή ότι μπορεί να γράψει την επόμενη εντολή.

Σε περίπτωση που η εντολή του επεξεργαστή απαιτεί απόκριση από την FPGA, είναι δηλαδή εντολή τύπου ανάγνωσης, τότε ο συγχρονισμός επιτυγχάνεται μέσω του bit Data Read Ready (DRR) του καταχωρητή SSR. Πιο συγκεκριμένα, όταν η FPGA δεχθεί μία εντολή ανάγνωσης, την εκτελεί, παράγει τα αποτελέσματα και τα γράφει στους καταχωρητές DR1,2. Μόλις γράψει τα δεδομένα, θέτει το bit DRR του SSR προκειμένου να ενημερώσει τον επεξεργαστή ότι τα δεδομένα που περιέχονται στους καταχωρητές DR είναι έγκυρα. Ο επεξεργαστής εφόσον δει το DRR να έχει τεθεί διαβάζει τα δεδομένα και μηδενίζει το DRR ώστε η FPGA να γνωρίζει με τη σειρά της ότι τα τελευταία δεδομένα που τοποθέτησε στους DR έχουν διαβαστεί.

Τέλος στον καταχωρητή SSR υπάρχει το bit Stop and Listen (SAL), το οποίο τείθεται από τον επεξεργαστή όταν αιτείται από την FPGA να διακόψει όλες τις υπόλοιπες λειτουργίες της και να τεθεί υπο τον πλήρη έλεγχο του. Αυτή η λειτουργία είναι χρήσιμη στις περιπτώσεις πραγματοποίησης αποσφαλμάτωσης (debugging) κατά την διάρκεια ανάπτυξης ή σε περιπτώσεις που απαιτείται επέμβαση λόγω βλαβών η κακής λειτουργίας.

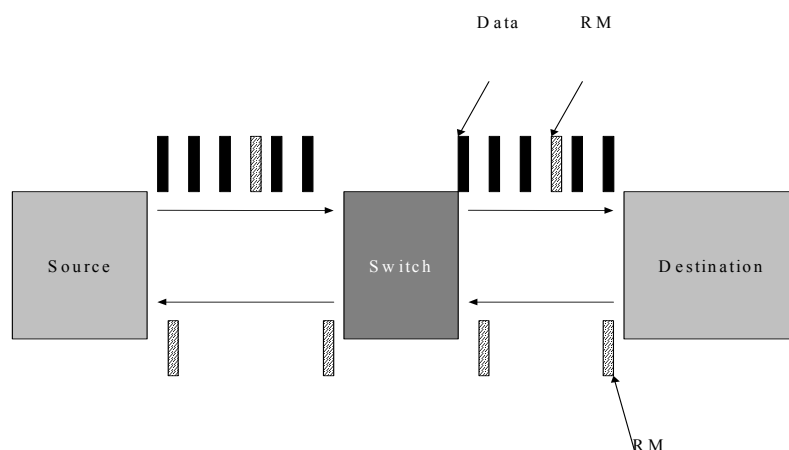
Η ανταλλαγή της πληροφορίας συγχρονισμού μεταξύ των δύο μερών της διεπαφής (επαξεργαστής-FPGA) πραγματοποιείται μέσω της αμοιβαίας πρόσβασης σε κοινόχρηστους πόρους (καταχωρητής SSR). Συνεπώς η ευθύνη της ενημέρωσης για την κατάσταση του συγχρονισμού ανήκει στο κάθε μέρος της διεπαφής ξεχωριστά. Από την μεριά του επεξεργαστή λοιπόν, το λογισμικό του θα πρέπει να υιοθετεί την λειτουργία του polling στον καταχωρητή SSR προκειμένου να ενημερώνεται για την κατάσταση του συγχρονισμού της διεπαφής.

## ***Παράρτημα Β - Έλεγχος Ροής στην κάρτα Εξυπηρετητή ABR***

## Γενικά για τον έλεγχο ροής ATM στην κίνηση τύπου ABR

### Γενικό μοντέλο Ελέγχου Ροής ABR

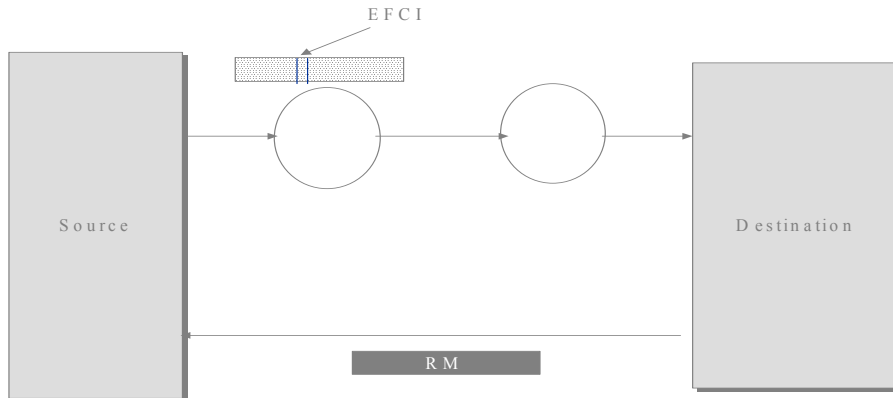
Οι ABR μηχανισμοί επιτρέπουν στο δίκτυο να διαμοιράζει το διαθέσιμο εύρος ζώνης δίκαια και αποτελεσματικά μεταξύ των ενεργών πηγών κυκλοφορίας. Με τις διαδικασίες Ελέγχου Ροής ABR, οι πηγές οδηγούνται στο να περιορίσουν τους ρυθμούς μετάδοσης δεδομένων σε τιμές που είναι επιτρεπτές από το δίκτυο. Το δίκτυο αποτελείται από μεταγωγείς (όπως ο Κόμβος Πρόσβασης) οι οποίοι χρησιμοποιούν την εσωτερική πληροφορία που αφορά τον φόρτο τους προκειμένου να υπολογίσουν τις επιτρεπόμενες τιμές για τις ταχύτητες μετάδοσης των πηγών. Αυτές οι τιμές αποστέλλονται προς τις πηγές ως ανάδραση μέσω ειδικών cells τα οποία ονομάζονται *Resource Management (RM)* cells. Τα RM cells παράγονται από τις πηγές και ταξιδεύουν κατα μήκος του μονοπατιού δεδομένων έως τα συστήματα προορισμού (destination end systems). Τα συστήματα προορισμού απλώς επιστρέφουν τα RM cells πίσω στις πηγές. Η διαδικασία του Ελέγχου Ροής και των στοιχείων που συμμετέχουν σε αυτή φαίνονται στο Σχήμα 1



**Σχήμα 1 Το μοντέλο Ελέγχου Ροής: Πηγή(Source), Μεταγωγός (Switch), Προορισμός (Destination)**

Σύμφωνα με το πρότυπο οι μεταγωγείς έχουν τρεις τρόπους να στείλουν ανάδραση προς τις πηγές.:

1. Η κεφαλίδα του κάθε cell περιέχει ένα bit το οποίο ονομάζεται Explicit Forward Congestion Indication (EFCI), το οποίο μπορεί να τεθεί από έναν μεταγωγό στον οποίο υφίσταται συμφόριση (congestion). Τέτοιοι μεταγωγείς ονομάζονται «δυαδικοί» (binary) ή «EFCI» μεταγωγείς. Τα συστήματα προορισμού στην συνέχεια συγκεντρώνουν την πληροφορία των EFCI bits και επιστρέφουν ανάδραση προς την πηγή μέσω ενός RM cell. Μία γενική μορφή της Δυαδικής Ανάδρασης (Binary Feedback) φαίνεται στο Σχήμα 2.



**Σχήμα 2 Γενικό Σχήμα Δυαδικής Ανάδρασης (Binary Feedback)**

2. Τα RM cells έχουν δύο bits στο payload τους, τα οποία ονομάζονται Congestion Indication (CI) bit και No Increase (NI) bit και τείθενται από μεταγωγείς στους οποίους υφίσταται συμφόρηση. Οι μεταγωγείς που χρησιμοποιούνται μόνον αυτόν τον μηχανισμό ονομάζονται Relative Rate Marking μεταγωγείς.
3. Τα RM cells επίσης έχουν ένα πεδίο στο payload τους το οποίο ονομάζεται Explicit Rate (ER) και μεταβάλλεται από μεταγωγείς στους οποίους υφίσταται συμφόρηση σε οποιαδήποτε επιθυμητή τιμή. Τέτοιοι μεταγωγείς ονομάζονται Explicit Rate μεταγωγείς.

Οι μεταγωγείς Explicit Rate σε κανονικές συνθήκες περιμένουν την άφιξη ενός RM cell για να παρέχουν ανάδραση στην πηγή. Ωστόσο, κάτω από έντονες καταστάσεις συμφόρησης, έχουν το δικαίωμα να παράγουν ένα RM cell και να το στείλουν αμέσως στην πηγή. Αυτός ο προαιρετικός μηχανισμός ονομάζεται Backward Explicit Congestion Notification (BECN).

### **ABR παράμετροι**

Την στιγμή της εγκατάστασης μίας σύνδεσης, η πηγές ABR διαπραγματεύονται με το δίκτυο διάφορες λειτουργικές παραμέτρους. Η πρώτη από αυτές είναι το Peak Cell Rate (PCR). Αυτός είναι ο μέγιστος ρυθμός με τον οποίο η πηγή επιτρέπεται να μεταδώσει κίνηση στην συγκεκριμένη σύνδεση (VC). Η πηγή μπορεί επίσης να αιτηθεί ένα Minimum Cell Rate (MCR) ο οποίος ο ελάχιστος εγγυημένος ρυθμός μετάδοσης για την συγκεκριμένη σύνδεση. Το δίκτυο υποχρεούται να εξασφαλίσει το απαιτούμενο εύρος ζώνης για την σύνδεση (VC) για την ικανοποίηση των παραμέτρων οι οποίοι ζητήθηκαν. Κατά την διάρκεια της μετάδοσης δεδομένων, ο επιτρεπόμενος ρυθμός μετάδοσης για την πηγή σε κάποια συγκεκριμένη χρονική στιγμή ονομάζεται Allowed Cell Rate (ACR). Ο ACR μεταβάλλεται δυναμικά μεταξύ του MCR και του PCR. Όταν η πηγή ξεκινά την μετάδοση ή έπειτα από μεγάλα ανενεργά (idle) διαστήματα, το ACR τείθεται σε μία αρχική τιμή η οποία ονομάζεται Initial Cell Rate (ICR).



Κατά την διάρκεια της ανάπτυξης του προτύπου RM, όλες οι αριθμητικές τιμές του αντικαταστάθηκαν από μνημονικά. Για παράδειγμα, αντί να πει «κάθε 32ο cell θα πρέπει να υπάρχει ένα RM cell» το πρότυπο αναφέρει «κάθε Nrm-οστό cell θα πρέπει να υπάρχει RM cell». Σε αυτή την περίπτωση, Nrm είναι μία παράμετρος η οποία καθορίζει κάθε πόσα cells θα πρέπει να αποστέλεται ένα RM cell και η προκαθορισμένη τιμή του είναι 32. Κάποιες από τις παραμέτρους είναι σταθερές ενώ άλλες είναι διαπραγματεύσιμες. Μία πλήρης λίστα των παραμέτρων που χρησιμοποιούνται στον μηχανισμό ABR παρουσιάζεται στο Σχήμα 3.

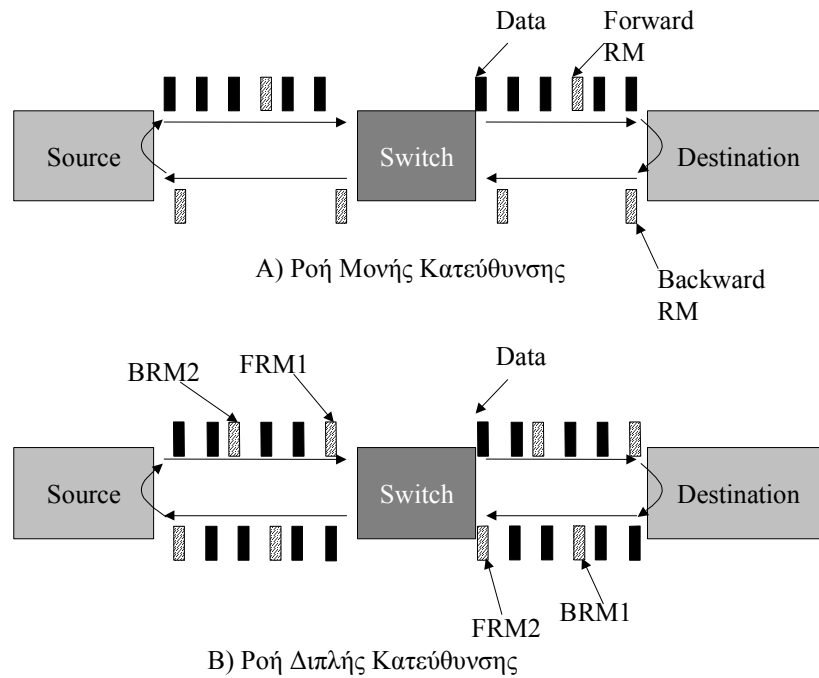
<b>Label</b>	<b>Expansion</b>	<b>5.3.1.2 Default Value</b>
PCR	Peak Cell Rate	-
MCR	Minimum Cell Rate	0
ACR	Allowed Cell Rate	-
ICR	Initial Cell Rate	PCR
TCR	Tagged Cell Rate	10 cells/s
Nrm	No of cells between RM	32
Mrm	Controls BW allocation between FRM, BRM & data cells	2
Trm	Upper Bound of Inter-FRM time	100ms
RIF	Rate Increase Factor	1/16
RDF	Rate Decrease Factor	1/16
ADTF	ACR Decrease Time factor	0.5ms
TBE	Transient Buffer Exposure	16.777.215
CRM	Missing RM cell count	TBE/Nrm
CDF	Cutoff Decrease factor	1/16
FRTT	Fixed Round Trip time	-

**Σχήμα 3** Λίστα των παραμέτρων ABR

### **Forward και Backward RM cells**

Τα RM cells που ταξιδεύουν από την πηγή στον προορισμό ονομάζονται Forward RM (FRM) cells. Ο προορισμός αντιστρέφει την πορεία αυτών RM cells και τα

επιστρέφει πίσω στην πηγή πάνω από το ίδιο VC. Αυτά τα RM cells που ταξιδεύουν από τον προορισμό προς την πηγή ονομάζονται Backward RM (BRM) cells. Τα Forward και Backward RM cells φαίνονται στο Σχήμα 4.



Σχήμα 4 Forward και Backward RM cells

Ας σημειωθεί ότι όταν υπάρχει κυκλοφορία και προς τις δύο κατευθύνσεις, υπάρχουν FRMs και BRMs και στις δύο κατευθύνσεις πάνω στο VC. Ένα bit στο payload του RM cell δείχνει αν είναι FRM ή BRM. Αυτό το bit κατεύθυνσης (DIR) αλλάζει από 0 σε 1 στον προορισμό.

### Δομή των RM cells

Η πλήρης δομή των RM cells φαίνεται στο Σχήμα 5.

ATM Header	5Bytes	
<b>Protocol ID</b>	1Byte	1=ABR
<b>Direction</b>	1bit	0=Forward
<b>Backward Notification</b>	1bit	1=Switch/Dest generated
<b>Congestion Indication</b>	1bit	1=High Congestion
<b>No Increase</b>	1bit	1=Mild Congestion
<b>Request Acknowledge</b>	1bit	
<b>Reserved</b>	3bits	
<b>Explicit Rate</b>	2Bytes	
<b>Current Cell Rate</b>	2Bytes	
<b>Minimum Cell Rate</b>	2Bytes	
<b>Queue Length</b>	4Bytes	
<b>Sequence Number</b>	4Bytes	
<b>Reserved</b>	30.75Bytes	
<b>CRC-10</b>	10bits	

Σχήμα 5 Τα πεδία του RM cell

Κάθε RM cell περιέχει τον συνηθισμένο ATM header που αποτελείται από 5 bytes. Το πεδίο Payload Type Indicator (PTI) έχει την τιμή 0b110 για να υποδεικίσει ότι το cell είναι ένα RM cell. Το πεδίο Protocol Id, το οποίο έχει μήκος 1 byte τείθεται στην τιμή ένα για να υποδεικίσει ότι πρόκειται για cell τύπου ABR. Το bit DIR χρησιμοποιείται για να διακρίνονται τα Forward από τα Backward RM cells. Το bit Backward Notification (BN) τείθεται μόνο σε cells BECN που παράγονται από τους μεταγωγείς. Το bit Congestion Indication (CI), χρησιμοποιείται σε Relative Rate Marking μεταγωγείς. Το bit No Increase (NI) είναι διαθέσιμο επίσης στους Relative Rate Marking μεταγωγείς και όταν τείθεται υποδεικνύει την ύπαρξη μέτριας συμφόρησης στον μεταγωγό. Τα πεδία Request/Acknowledge, Queue Length και Sequence Number υπάρχουν για λόγους συμβατότητας με παλαιότερα πρότυπα και δεν χρησιμοποιούνται από το ATM Forum.

Το πεδίο Current Cell Rate (CCR) χρησιμοποιείται από την πηγή για να παρέχει στο δίκτυο την τιμή του τρέχοντος ρυθμού μετάδοσης. Το πεδίο Minimum Cell Rate (MCR) είναι πλεονάζον από την άποψη ότι όπως και τα PCR, ICR και άλλοι παράμετροι δεν μεταβάλλεται κατά την διάρκεια ζωής μίας σύνδεσης. Ωστόσο, η

ύπαρξη τους στα RM cells ελατώνει τον αριθμό των ανακτήσεων πληροφορίας που απαιτούνται στον μεταγωγό.

Τα πεδία ER, CI και NI χρησιμοποιούνται από το δίκτυο για την παροχή ανάδρασης προς τις πηγές. Το πεδίο ER περιέχει τον μέγιστο ρυθμό μετάδοσης της πηγής ο οποίος είναι επιτρεπτός από το δίκτυο. Όταν υπάρχουν πολλοί μεταγωγοί κατά μήκος του μονοπατιού της σύνδεσης, η ανάδραση που φτάνει στην πηγή καταλήγει να είναι αυτή που παράγεται από τον κόμβο με την μεγαλύτερη συμφόρηση.

Όλοι οι ρυθμοί που βρίσκονται στα πεδία του RM cell αναπαριστώνται χρησιμοποιώντας ένα ειδικό format κινητής υποδιαστολής 16-bit, το οποίο επιτρέπει μέγιστη τιμή ίση με 4.290.772.992 cells/sec (1.8 terabits per second). Ωστόσο κατά την διάρκεια της εγκατάστασης μίας σύνδεσης, οι παράμετροι του ρυθμού μετάδοσης διαπραγματεύονται χρησιμοποιώντας ένα ακέραιο format 24-bit, το οποίο περιορίζει την μέγιστη τιμή στα 16.777.215 cells/sec ή 7.1 Gb/s.

## Συμπεριφορά Μεταγωγών στον Έλεγχο Ροής

Το πρότυπο του ATM Forum για τον Έλεγχο Ροής προσδιορίζει ότι οι μεταγωγείς πρέπει να υλοποιούν κάποιο μοντέλο ελέγχου συμφόρησης καθώς και κανόνες που αφορούν την επεξεργασία, την διαχείριση ουρών και την παραγωγή RM cells.

- **Κανόνες Μεταγωγών 1:** Αυτός ο κανόνας προσδιορίζει ότι μία ή περισσότερες μέθοδοι ανάδρασης ή marking θα πρέπει να υλοποιηθούν στον μεταγωγό. Οι δυνατές μέθοδοι είναι:
  - **EFCI Marking:** Προσδιορίζει την δυαδική (bit-based) μέθοδο ανάδρασης, όπου οι μεταγωγοί θέτουν το bit EFCI στους headers των cells δεδομένων. Οι προορισμοί διατηρούν την κατάσταση του EFCI για κάθε VC και θέτουν το CI bit στα backward RM cells αν το EFCI bit κάποιου VC έχει τεθεί. Ας σημειωθεί ότι η κατάσταση του EFCI ενός VC στον προορισμό μεταβαίνει στην κατάσταση set ή reset οποτεδήποτε ένα εισερχόμενο cell έχει το EFCI του σε κατάσταση set ή reset αντίστοιχα.
  - **Relative Rate Marking:** Αυτή η λειτουργία επιτρέπει στον μεταγωγέα να θέσει δύο bits του RM cell τα οποία έχουν μία συγκεκριμένη σημασία για τις πηγές που τα δέχονται. Το bit CI όταν είναι σε κατάσταση set ζητάει από την πηγή να ελατώσει τον τρέχοντα ρυθμό μετάδοσης ACR ενώ το bit NI ζητάει απλώς να μην τον αυξήσει. Είναι άξιο παρατήρησης ότι ο ρυθμός μετάδοσης της πηγής μπορεί να μειωθεί περαιτέρω χρησιμοποιώντας το πεδίο Explicit Rate. Αυτά τα δύο bits επιτρέπουν στους μεταγωγούς μεγαλύτερη ευελιξία από ότι το bit EFCI. Πιο συγκεκριμένα, οι μεταγωγοί μπορούν να αποφύγουν με αυτήν την τεχνική ένα πρόβλημα που είναι γνωστό ως «beat-down fairness» το οποίο εμφανίζεται στα σενάρια EFCI. Αυτό το πρόβλημα εμφανίζεται εξαιτίας του ότι οι συνδέσεις που περνούν από μεγάλο αριθμό κόμβων έχουν μεγαλύτερη πιθανότητα να έχουν το bit EFCI σε κατάσταση set, σε σχέση με τις συνδέσεις που περνούν από μικρότερο αριθμό κόμβων.

- **Explicit Rate Marking:** Αυτή η μέθοδος δίνει την δυνατότητα σε έναν μεταγωγό να προσδιορίσει με ακρίβεια τον ρυθμό στον οποίο θέλει να μεταδίδει η πηγή. Για να εξασφαλιστεί η συνεργασία μεταξύ των πολλών μεταγωγών που αποτελούν το μονοπάτι μίας σύνδεση, ο κάθε ένας από αυτούς μπορεί μόνο να ελλατώσει (αλλά όχι να αυξήσει) το πεδίο ER των RM cells.
- **VS/VD Control:** Με αυτή τη μέθοδο, ο μεταγωγός μπορεί να χωρίσει τον ABR βρόχο ελέγχου σε ξεχωριστά κομμάτια εμφανίζοντας τον εαυτό του ως μία νοητή πηγή (virtual source) στην μία πλευρά του βρόχου και ως έναν νοητό προορισμό (virtual destination) στην άλλη πλευρά.
- **Κανόνες Μεταγωγών 2:** Αυτός ο κανόνας καθορίζει το πως ένας μεταγωγός θα παράγει ένα RM cell στην περίπτωση που υφίσταται μεγάλη συμφόρηση αλλά αργεί να δεχτεί ένα RM cell από την πηγή. Βασικά, ο κανόνας επιτρέπει σε τέτοια RM cells, να ελλατώνουν μόνο τον ρυθμό της πηγής.
- **Κανόνες Μεταγωγών 3:** Αυτός ο κανόνας λέει ότι τα RM cells μπορούν να μεταδωθούν από έναν μεταγωγό με διαφορετική σειρά σε σχέση με τα cells δεδομένων, από αυτή με την οποία εισήλθαν, αλλά δεν επιτρέπεται να αλλαχθεί η συνέχεια και η σειρά μεταξύ τους. Αυτός ο κανόνας επιτρέπει στον μεταγωγό να τοποθετεί ένα RM cell σε μία ουρά προτεραιότητας με σκοπό την γρηγορότερη ανάδραση προς την πηγή.
- **Κανόνες Μεταγωγών 4 και 5:** Ο κανόνας 4 προσδιορίζει τη συσχέτιση με το I.371 draft της ITU-T και εξασφαλίζει την ακεραιότητα του πεδίου MCR στο εσωτερικό του RM cell. Ο κανόνας 5 επιτρέπει την προεραϊκή υλοποίηση της πολιτικής use-it-or-lose-it στον μεταγωγό.

### *Αλγόριθμος Ελέγχου Ροής στην Κάρτα Εξυπηρετητή ABR*

#### **Τεχνική UCSC**

Αυτή τεχνική προτάθηκε από του ερευνητές Kalamroukas και Varma του Πανεπιστημίου της Καλιφόρνια και Ramakrishnan της AT&T Research.

#### **Ο βασικός αλγόριθμος**

Η συγκεκριμένη τεχνική προσεγγίζει έξυπνα την τεχνική MIT πετυχαίνοντας αποθήκευση πληροφορίας της τάξης  $O(1)$  μειώνοντας έτσι την υπολογιστική πολυπλοκότητα από  $O(N)$  σε  $O(1)$ . Επιπρόσθετα η τεχνική διαμοιράζει την  $O(N)$  επαναληπτικότητα σε συνεχόμενα RM cells. Αυτό έχει ως αποτέλεσμα, να πραγματοποιείται ανταλλαγή μεταξύ του χρόνου σύγκλισης και των απαιτήσεων σε αποθηκευτικό χώρο και της υπολογιστικής πολυπλοκότητας. Η χωρικές απαιτήσεις είναι της τάξης  $O(N)$  καθώς η τεχνική διατηρεί κάποια πληροφορία κατάστασης αν VC. Σε εξαιρετικές περιπτώσεις, είναι δυνατόν να επιτευχθεί βελτιστοποίηση χρησιμοποιώντας λειτουργίες ολίσθησης αντί των αντιστοίχων λειτουργιών πολλαπλασιασμού διαίρεσης.

Ο βασικός αλγόριθμος της τεχνικής είναι ο ακόλουθος. Η τεχνική θεωρεί ότι η πηγή ζητά έναν συγκεκριμένο ρυθμό και ο μεταγωγός προσπαθεί να ικανοποιήσει την ζήτηση. Στην τεχνική αυτή ένα  $VC_i$  αιτείται bandwidth ίσο με το  $\min(ER_i, CCR_i)$  ( $ER$ :Explicit Rate,  $CCR$ :Current Cell Rate). Μπορούμε να ονομάσουμε αυτή την ποσότητα «αίτηση» ενός  $VC_i$ . Η ίδια ποσότητα μπορεί επίσης να θεωρηθεί ως η τρέχουσα χρήση Bandwidth το  $VC$ . Η τεχνική υπολογίζει ένα «maximum bandwidth»  $A_{max}$  το οποίο εξαρτάται από την τωρινή κατάσταση του  $VC$ . Το  $A_{max}$  είναι είναι η δίκαιη κατανομή εύρους (fairshare) η οποία παρέχεται στην πηγή ως ανάδραση.

Στον συγκεκριμένο αλγόριθμο κάθε  $VC_i$  μπορεί να βρίσκεται σε μία από τις επόμενες δύο καταστάσεις:

1. **Bottlenecked:** σε αυτή την κατάσταση ο μεταγωγός δεν μπορεί να δεσμεύσει το αιτούμενο εύρος ζώνης από το  $VC_i$  στο link εξόδου. Σε αυτή την περίπτωση ισχύει  $A_{max} < \min(ER_i, CCR_i)$ . Το σύνολο των bottlenecked συνδέσεων ονομάζεται  $B$ . Με άλλα λόγια, οι bottlenecked συνδέσεις είναι αυτές που μπορούν να χρησιμοποιήσουν περισσότερο εύρος ζώνης του μεταγωγού.
2. **Satisfied:** σε αυτήν την περίπτωση ο μεταγωγός μπορεί να ικανοποιήσει την αίτηση και ισχύει  $A_{max} \geq \min(ER_i, CCR_i)$ . Το σύνολο των satisfied συνδέσεων ονομάζεται  $S$ . Με άλλα λόγια, οι satisfied συνδέσεις είναι εκείνες που δεν μπορούν να χρησιμοποιήσουν ούτε καν το τρέχον δεσμευμένο για αυτές εύρος ζώνης  $A_{max}$ . Υπο κάποια έννοια μπορούν να θεωρηθούν «κορεσμένες»

Τυπικά ένα δεδομένο  $VC_i$  θα βρίσκεται σε διαφορετικές καταστάσεις από μεταγωγό σε μεταγωγό. Ας σημειωθεί ότι οι συνδέσεις μπορούν να εναλλάσσονται από την μία κατάσταση στην άλλη ανάλογα με τις αιτήσεις τους και το διαθέσιμο εύρος ζώνης. Το ελεύθερο εύρος ζώνης (free bandwidth –  $Bf$ ) ορίζεται ως η ποσότητα του εύρους που είναι διαθέσιμο ως αποτέλεσμα του γεγονότος ότι οι satisfied συνδέσεις δεν χρησιμοποιούν το μερίδιο bandwidth που τους αναλογεί (equal share –  $Beq$ ). Ο υπολογισμός του  $A_{max}$  για μία σύνδεση πραγματοποιείται ως ακολούθως:

Πρώτα, ανιχνεύονται οι αλλαγές κατάστασης της σύνδεσης και ανανεώνονται οι διάφοροι παράμετροι:

- Αν  $A_{max} < \min(ER_i, CCR_i)$ , η σύνδεση μαρκάρεται ως “bottlenecked”. Σε αυτή την περίπτωση, εφόσον η προηγούμενη κατάσταση της σύνδεσης ήταν “satisfied”, ανανεώνεται η παράμετρος  $Bf$  (free bandwidth), και αυξάνεται κατά ένα ο αριθμός των “bottlenecked” συνδέσεων  $Nbot$ .
- Αν  $A_{max} \geq \min(ER_i, CCR_i)$  η σύνδεση μαρκάρεται ως “satisfied”, η τρέχουσα αιτούμενη δέσμευση  $Ai$  τίθεται ίση με το  $\min(ER_i, CCR_i)$  και ανανεώνεται η παράμετρος  $Bf$  (free bandwidth). Επιπλέον αν η προηγούμενη κατάσταση της σύνδεσης ήταν “bottlenecked”, ελατώνεται κατά ένα και ο ο αριθμός των “bottlenecked” συνδέσεων  $Nbot$ .

Το επόμενο βήμα είναι ο υπολογισμός της κατανομής εύρους ζώνης  $A_{max}$ :

Εάν μία σύνδεση  $VC_i \in B$ , δηλαδή η τωρινή της κατάσταση είναι “bottlenecked”, η νέα δίκαια κατανομή (fairshare)  $A_{max}$  υπολογίζεται ως εξής:

$$A_{\max} = B_{eq} + \frac{B_f}{N_{bot}}$$

Στην άλλη περίπτωση, εάν μία σύνδεση  $VC_i \in S$ , δηλαδή η τωρινή της κατάσταση είναι “satisfied”, αυτή την χειρίζεται ο αλγόριθμος ως “bottlenecked” και η νέα δίκαια κατανομή (fairshare)  $A_{\max}$  υπολογίζεται ως εξής:

$$A_{\max} = B_{eq} + \frac{B_f + A_i - B_{eq}}{N_{bot} + 1}$$

Στην προηγούμενη εξίσωση, παρατηρείται ότι η δέσμευση bandwidth της σύνδεσης πάνω από το δίκαιη κατανομή (equal share) της,  $A_i - B_{eq}$ , θεωρείται επίσης μέρος του ελεύθερου εύρους ζώνης (free bandwidth). Η χρήση του  $N_{bot} + 1$  στον παρονομαστή του κλάσματος φανερώνει ότι η πηγή θεωρείται για τον υπολογισμό ως “bottlenecked” σύνδεση. Ο σκοπός αυτού του υπολογισμού είναι να εξασφαλίσει ότι η κατανομή εύρους στις “satisfied” συνδέσεις είναι πάντα μικρότερη ή ίση σε σχέση με την κατανομή στις “bottlenecked” συνδέσεις. Με αυτόν τον τρόπο ο αλγόριθμος πέρνει πίσω το επιπλέον εύρος που παρείχε σε προηγούμενη χρονική στιγμή στην σύνδεση.

Τέλος η νέα δίκαια κατανομή (fairshare)  $A_{\max}$ , επιστρέφεται ως ανάδραση στην πηγή, μέσω της ανανέωσης του πεδίου explicit rate των RM cells της σύνδεσης ως ακολούθως:

$$ER_i = \min(ER_i, A_{\max})$$

## Χαρακτηριστικά του αλγορίθμου

Ο αλγόριθμος χαρακτηρίζεται από τους εμπνευστές του ως αλγόριθμος «state-maintaining» καθώς διατηρεί πληροφορία κατάστασης για ακάθε σύνδεση. Παρατηρείται ότι οι αλγόριθμοι που δεν διατηρούν πληροφορία κατάστασης ανά σύνδεση (stateless) είναι δυνατόν να δεσμεύσουν εύρος ζώνης τέτοιο ώστε να υπάρχουν σημαντικές ανακολουθίες μεταξύ του αθροίσματος των τιμών ER που παρέχονται ως ανάδραση στις συνδέσεις ABR και στο πραγματικό διαθέσιμο εύρος ζώνης.

Τα θετικά στοιχεία του συγκεκριμένου αλγορίθμου είναι τα ακόλουθα:

- Προσομοιώνεται ο αλγόριθμος MIT με πολυπλοκότητα μόλις  $O(1)$
- Scalability. Αν τα ανοιχτά VCs είναι πάντα ενεργά, τότε ο αλγόριθμος έχει υπολογιστική πολυπλοκότητα  $O(1)$  ανεξάρτητα από τον αριθμό των VCs.
- Ο αλγόριθμος δεν απαιτεί την αρχικοποίηση παραμέτρων.
- Είναι δυνατή η ανάλυση της απόδοσης είτε η ABR χωρητικότητα είναι σταθερή είτε είναι μεταβλητή.

Τα αρνητικά στοιχεία του αλγορίθμου είναι:

- Δεν μετράται το φορτίο (συνολικός ρυθμός εισόδου) του μεταγωγού. Ως αποτέλεσμα, αν μία πηγή στέλνει με ρυθμό κάτω από το CCR τότε το σημείο bottleneck θα υποχρησιμοποιείται.
- Δεν παρατηρείται η καθυστέρηση στις ουρές του μεταγωγού. Έτσι λάθη στην εκτίμηση της ABR χωρητικότητας μπορεί να έχουν ως αποτέλεσμα λάθη στην ανάδραση προς τις πηγές και συνεπώς την δημιουργία ουρών. Με αυτόν τον τρόπο υπάρχει η πιθανότητα δημιουργίας ουρών εάν δεν ληφθεί ως μετρικό η καθυστέρηση στις ουρές.
- Ο αλγόριθμος θεωρεί ότι το άθροισμα του αριθμού των “bottlenecked” και “satisfied” συνδέσεων είναι ίσο με το σύνολο των εγκατεστημένων συνδέσεων. Δεν μετρά όμως τις ενεργές συνδέσεις. Ως αποτέλεσμα, εάν μία σύνδεση είναι εγκατεστημένη, αλλά παραμένει ανενεργή για κάποιο χρονικό διάστημα, η παροχή εύρους στις υπόλοιπες συνδέσεις παραμένει χαμηλή και γεγονός που οδηγεί σε υποχρησιμοποίηση των πόρων.



## Βιβλιογραφία

- (1) ATM Forum, “UTOPIA Specification Level 1, Version 2.01”, af-phy-0017.000, 1994.
- (2) ATM Forum, “ATM User-Network Interface (UNI) Signalling Specification Version 4.0”, af-sig-0061.000, 1996.
- (3) ATM Forum, “Integrated Local Management Interface (ILMI) Specification Version 4.0”, af-ilmi-0065.000, 1996
- (4) ATM Forum, “Traffic Management Specification”, Version 4.1, af-tm-0121.000, March 1999.
- (5) Cisco Systems, “ATM Internetworking”, 1995
- (6) ITU-T, Recommendation I.363, “B-ISDN ATM adaptation layer specification”, Geneva 1997.
- (7) Motorola Inc., “MC92501 ATM Cell Processor User’s Manual”, 1998.
- (8) Motorola Inc., “MPC860 PowerQUICC User’s Manual”, 1998.
- (9) TranSwitch Corporation, “Cubit Pro Device CellBus Bus Switch Data Sheet”, 1997.
- (10) WindRiver Systems, “VxWorks Programmers Guide”, 1999.
- (11) Kalyanaraman S., “Traffic Management for the Available Bit Rate (ABR) Service in Asynchronous Transfer Mode (ATM) Networks”, PhD dissertation, 1997.
- (12) Kalampoukas L, Varma A. and Ramakrishnan K., “An efficient rate allocation algorithm for ATM networks providing max-min fairness”, in Proc of 6<sup>th</sup> HPN, pp. 143-155, 1995
- (13) Συννεφάκης Γ. “Δίκτυα Διασύνδεσης σε Συστήματα Μεταγωγέων Ασύγχρονου Τρόπου Μεταφοράς”, Διδακτορική Διατριβή, 2001
- (14) Kernighan W.B., Ritchie M.D., “The C Programming Language (Second Edition)”, 1990