

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΡΟΣΑΡΜΟΓΗ ΛΕΙΤΟΥΡΓΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΚΑΙ ΜΕΤΑΤΡΟΠΕΣ
ΥΛΙΚΟΥ ΣΕ ΠΛΑΚΕΤΑ ΕΝΣΩΜΑΤΩΜΕΝΟΥ ΕΠΕΞΕΡΓΑΣΤΗ

ΔΗΜΗΤΡΙΟΣ Α. ΜΕΪΝΤΑΝΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΗΡΑΚΛΕΙΟ, ΙΟΥΝΙΟΣ 2002

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΡΟΣΑΡΜΟΓΗ ΛΕΙΤΟΥΡΓΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΚΑΙ ΜΕΤΑΤΡΟΠΕΣ ΥΛΙΚΟΥ ΣΕ ΠΛΑΚΕΤΑ

ΕΝΣΩΜΑΤΩΜΕΝΟΥ ΕΠΕΞΕΡΓΑΣΤΗ

Εργασία που υποβλήθηκε από τον
ΔΗΜΗΤΡΙΟ Α. ΜΕΪΝΤΑΝΗ
ως μερική εκπλήρωση των απαιτήσεων
για την απόκτηση
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

Δημήτριος Α. Μεϊντάνης
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Εξεταστική
Επιτροπή:

Μανόλης Κατεβαίνης, Καθηγητής,
Επόπτης

Ευάγγελος Μαρκάτος, Αναπληρωτής Καθηγητής,
Μέλος

Απόστολος Τραγανίτης, Αναπληρωτής Καθηγητής,
Μέλος

Δεκτή:

Πάνος Κωνσταντόπουλος, Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Οκτώβριος 2002

ΠΡΟΣΑΡΜΟΓΗ ΛΕΙΤΟΥΡΓΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ
ΚΑΙ ΜΕΤΑΤΡΟΠΕΣ ΥΛΙΚΟΥ ΣΕ ΠΛΑΚΕΤΑ ΕΝΣΩΜΑΤΩΜΕΝΟΥ
ΕΠΕΞΕΡΓΑΣΤΗ

Δημήτριος Α. Μεϊντάνης

Μεταπτυχιακή Εργασία

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Περίληψη

Η έρευνα που διεξήχθη για την εργασία αυτή, ήταν ένα μέρος απο τη συνολική προσπάθεια για την κατασκευή ενός φτηνού, συμπαγή αλλά εν τούτοις πολυδύναμου Residential Gateway. Όπως γνωρίζουμε, στα περισσότερα συμβατικά σπίτια, ηλεκτρικά και ηλεκτρονικά συστήματα λειτουργούν αρκετά ανεξάρτητα. Αυτά τα συστήματα συμπεριλαμβάνουν (αλλά δεν είναι περιορισμένα σε) τον φωτισμό, την ασφάλεια, την ψυχαγωγία, τα τηλέφωνα, τον έλεγχο κλιματισμού και τους ηλεκτρονικούς υπολογιστές. Η ψηφιακή επανάσταση και οι νέες υπηρεσίες πληροφορίας έχουν δημιουργήσει πολλές νέες εφαρμογές που εκμεταλλεύονται το "δικτυωμένο" σπίτι.

Η αδυσώπητη μετατροπή όλων των μέσων σε ψηφιακή πληροφορία, και η πανταχού παρουσία του διαδικτύου, έχουν δώσει αφορμή για το "Web lifestyle". Υψηλής ταχύτητας πρόσβαση στο διαδίκτυο, υπόσχεται την ξεπέραση των περιορισμών του bandwidth, την δημιουργία νέων δυνατοτήτων για επικοινωνία και οικιακή ψυχαγωγία. Η ανάγκη για να μοιράσουμε την συνεχόμενη αύξηση της πληροφορίας οπουδήποτε στο σπίτι, όλο και μεγαλώνει. Αυτοί οι παράγοντες έχουν δημιουργήσει την ανάγκη για δίκτυα που είναι ραμένα για την οικιακή αγορά.

Στην μεταπτυχιακή αυτή εργασία, προτείνουμε και σχεδιάζουμε ένα προτότυπο Residential Gateway, που υποστηρίζει τρεις γραμμές ISDN, κάρτες επέκτασης PCMCIA, USB, και τρία Ethernet Interfaces, ένα από τα οποία είναι optical. Πρόκειται για ένα χαμηλού κόστους, αλλά παρ' όλα αυτά πολύ ισχυρού, RG που στοχεύει στην αγορά των μικρομεσαίων επιχειρήσεων. Πλήρως ευέλικτο στο να υποστηρίζει νέες τεχνολογίες, μιας και έχει εγκατα-

στημένο το λειτουργικό linux, έναν open source λειτουργικό, μπορεί να ικανοποιήσει μία επιχείρηση με υψηλές απαιτήσεις δικτύωσης.

Μελετάμε τα χαρακτηριστικά και τον τρόπο λειτουργίας του επεξεργαστή MPC 8260 της Motorola. Ένας επεξεργαστής με ενσωματωμένο έναν RISC Communication Processor, ο οποίος αναλαμβάνει να εκτελέσει όλες τις λειτουργίες επικοινωνίας με το εξωτερικό περιβάλλον. Ο μικρός αυτός ενσωματωμένος επεξεργαστής μπορεί να υποστηρίξει έως και δύο Utopia Level 2 Interfaces, τρεις 100Mbps Ethernet Interfaces καθώς και δύο 11500bps Serial Interfaces.

Επίπρόσθετα, μελετάμε το λειτουργικό σύστημα linux 2.4.7, και το τροποποιούμε κατάλληλα έτσι ώστε να εκτελείται ομαλά στο σύστημά μας.

Τέλος, στόχος της εργασίας αυτής είναι η ανάπτυξη της διασύνδεσης του ATM ενός καινούριου κατασκευασμένου Residential Gateway. Η διασύνδεση αυτή είναι ένας linux device driver που υποστηρίζει έως και 2048 UBR και ABR ATM κανάλια και έως οκτώ ουρές προτεραιότητας. Επίσης στον driver αυτό υλοποιείται το linux ATM interface που ορίζεται στα specifications του linux ATM.

Επόπτης: **Μανόλης Κατεβαίνης**, Καθηγητής Πανεπιστημίου Κρήτης, Τμήμα Επιστήμης Υπολογιστών

HARDWARE CONVERSION AND SOFTWARE
IMPORT TO A PROTOTYPE MICROPROCESSOR
BOARD

Dimitrios A. Meidanis

Master Thesis

UNIVERSITY OF CRETE
DEPARTMENT OF COMPUTER SCIENCE

Abstract

The research conducted for this thesis was part of an overall effort to develop a cheap, compact and yet powerful Residential Gateway. As we know, in most conventional homes, electrical and electronic systems operate fairly independently. These systems include (but aren't limited to) lighting, security, entertainment, phones, climate control and PCs. The digital revolution and new information services have created many new applications that take advantage of the "wired" home.

Relentless conversion to all-digital media and the ubiquity of the Internet have given rise to the "Web lifestyle". High-speed Internet access promises to overcome bandwidth limitations, opening new possibilities for communications and home entertainment. The need to share ever-increasing amounts of information anywhere in the household is growing. These factors have created a need for networks tailored to the residential market.

In this master thesis, we propose and design a new prototype Residential Gateway, which has three ISDN interfaces, two slots for PCMCIA expansion cards, USB interface, and three Ethernet Interfaces, one of which is optical. It is a compact, affordable but yet powerful RG, that points the medium size companies market. Fully versatile to support new technologies, due to the fact that linux kernel runs on the system board, the RG can satisfy any company with high network restrictions.

We study the characteristics and the operation of the MPC 8260 processor by Motorola. A processor with an embedded RISC communication processor, who is responsible for the

communication control with the external devices. This RISC processor, is very powerfull, and can support up to two Utopia Level 2 interfaces, three 100Mbps Ethernet interfaces, as well as up to two 15200 bps Serial Interfaces.

In addition, we study the operating system linux 2.4.7, and we modify it so that it runs smoothly to our custom board.

Finally, the goal of this thesis is to develop the ATM interface of the new constructed board. This interface is a linux device driver that can support up to 2048 ABR and UBR ATM channels, and up to eight priority queues. In this driver, we also implement the linux ATM interface, that is specified by the linux ATM society.

Περιεχόμενα

1	Εισαγωγή	8
1.1	Υπόβαθρο	8
1.2	Γιατί Τώρα;	9
1.3	Standards	9
1.4	Χρειαζόμαστε πραγματικά ένα Home Gateway;	9
2	Το περιεχόμενο της εργασίας	11
2.0.1	MPC8260 Manual Study	11
2.0.2	Boot Loader Study	11
2.0.3	Εισαγωγή του Boot Loader	12
2.0.4	Linux Kernel Import	12
2.0.5	Η δημιουργία του Infostar C3	13
2.0.6	Motorola VADS development board	13
2.0.7	Μελέτη του Motorola ATM Interface	14
2.0.8	Η δημιουργία του ATM driver	14
3	To Residential Gateway	15
3.1	Επισκόπηση των Residential Gateway	15
3.2	Η Αρχιτεκτονική των Residential Gateway	16
3.3	The Open Service Gateway Initiative	18
3.4	Προτερήματα	19
3.5	Προκλήσεις	20
3.6	Υπάρχοντα Residential Gateways	21
4	InfoStar C3	22
4.1	Εισαγωγή	22
4.2	Αρχιτεκτονική	24

4.3	Η δημιουργία του σχηματικού	24
4.4	Η δημιουργία του τυπωμένου κυκλώματος	26
5	Motorola MPC8260 Processor	29
5.1	Ανασκόπηση της Αρχιτεκτονικής	29
5.2	MPC603e Core	29
5.3	System Interface Unit (SIU)	31
5.4	Μονάδα Επεξεργαστή Επικοινωνιών (CPM)	31
6	Monitor Program	33
6.1	Ανασκόπηση του InfoStar C1	33
6.2	Ανασκόπηση του Monitor προγράμματος	34
6.3	Διαδικασία Import	36
6.3.1	PPCBoot 1.0.1 loader	37
6.3.2	Flash Driver	38
6.4	Ethernet Support Configuration	39
7	Linux Kernel	40
7.1	Linux Kernel Version	40
7.2	Debugging method	40
8	ATM Interface	42
8.1	Επικοινωνιακή Μονάδα Επεξεργαστή	42
8.1.1	Χαρακτηριστικά	42
8.2	Διαχειριστής Επικοινωνιών (CP)	44
8.2.1	Χαρακτηριστικά	44
8.2.2	CP Block Diagram	44
8.3	Ανασκόπηση του ελεγκτή του ATM	45
8.3.1	Ανασκόπηση του Transmitter	45
8.3.2	Ανασκόπηση του AAL5 Transmitter	46
8.3.3	Ανασκόπηση του Receiver	46
8.3.4	Ανασκόπηση του AAL5 Receiver	47
8.3.5	ATM Pace Control	47
8.3.6	Address Lookup Mechanism	48
8.3.7	Δομή μνήμης του ATM	50
8.3.8	ATM Exceptions	53
8.3.9	Utopia Interface	55
8.4	ATM Linux Driver	55

8.4.1 Ανασκόπηση του Driver	55
8.4.2 Η δομή του driver	57
8.4.3 Driver Interface	59
8.5 Physical Custom Driver	60
9 Συμπεράσματα	63

Εισαγωγή

Ένα Residential Gateway συνδέει το local area network (LAN) του σπιτιού με το Διαδίκτυο. Μια συσκευή υλικού παρόμοια στην εμφάνιση με έναν δρομολογητή, το Residential Gateway παρέχει έναν μοναδικό συνδυασμό χαρακτηριστικών γνωρισμάτων που ενδιαφέρει πολλές οικογένειες.

1.1 Υπόβαθρο

Εξ ορισμού, μια πύλη ενώνει δύο δίκτυα μαζί. Η λέξη είναι ένα μέρος του λεξικού για τα δίκτυα εδώ και τριάντα έτη, και τα Gateway έχουν διαδραματίσει ιστορικά έναν βασικό ρόλο στην ανάπτυξη του Διαδικτύου. Τα Gateway υπάρχουν σε πολλαπλές μορφές συμπεριλαμβανομένων των γενικής χρήσης κεντρικών υπολογιστών με τους πολλαπλούς προσαρμοστές δικτύων (που είναι γνωστοί επίσης ως multi-homed υπολογιστές) και τους δρομολογητές.

Οι παραδοσιακοί Gateway έχουν εγκατασταθεί στα δωμάτια ή τα ντουλάπια κεντρικών υπολογιστών, αλλά τα Residential Gateway φέρνουν αυτές τις συσκευές στο σπίτι. Τα Residential Gateway ποικίλλουν σημαντικά στις δυνατότητες κάλυψής τους, έτσι ώστε κανένα "τυπικό" Residential Gateway να μην υπάρχει. Εντούτοις, τα περισσότερα Residential Gateway υποστηρίζουν τα ακόλουθα βασικά χαρακτηριστικά γνωρίσματα :

- Broadband (often DSL) service connectivity
- Internet connection sharing
- Firewall security

1.2 Γιατί Τώρα;

Δεδομένου ότι η βασική δικτύωση συνεχίζει να αυξάνεται στη δημοτικότητα και την δυνατότητα κάλυψης, μερικοί προμηθευτές προϊόντων βλέπουν μια ευκαιρία να καταστήσουν τα βασικά δίκτυα ευκολότερα να χτίσουν και να χρησιμοποιηθούν. Πολλά στοιχεία ενός βασικού οικογενειακού δικτύου, όπως το IP addressing , παρουσιάζουν περιττή πολυπλοκότητα στο μη τεχνικό ιδιοκτήτη σπιτιού. Οι προμηθευτές δεν κατασκευάζουν εσκεμμένα προϊόντα που να είναι δύσκολη η χρήση τους, αλλά οι core network τεχνολογίες, ιδιαίτερα Ethernet.

Η εμφάνιση της broadband δικτύωσης προκαλεί επίσης μερικούς να θεωρήσουν ότι ο χρόνος έχει έρθει για τα Residential Gateway. Το broadband υπόσχεται να φέρει μια νέα πληθώρα εφαρμογών, βασισμένων στο WEB, στο σπίτι συμπεριλαμβανομένης της σε πραγματικό χρόνο, υψηλής ποιότητας ροής ήχου και βίντεο και του αληθινά immersive online παιχνιδιού. Πολλοί θεωρούν ότι μια κεντρική συσκευή όπως ένα Residential Gateway θα είναι σε θέση να υποστηρίξει καλύτερα αυτήν την νέα γενιά των τεχνολογιών του Web.

Επίσης, οι προμηθευτές προϊόντων ελπίζουν ότι τα Residential Gateway θα διαδραματίσουν βασικό ρόλο στην αύξηση του οικιακού αυτοματισμού. Αντί μόνο να συνδέει με το Διαδίκτυο, ένα Residential Gateway επίσης θα αλληλεπιδρά με το τηλέφωνο, το στερεοφωνικό, το σύστημα θέρμανσης, αερισμού και κλιματισμού, τις συσκευές κουζινών, και άλλες συσκευές network-enabled του μέλλοντος.

1.3 Standards

Επειδή τα προϊόντα Gateway παραμένουν σχετικά ανώριμα, τα τεχνικά πρότυπα τους συνεχίζουν να εξελίσσονται. Οι σχετικές προσπάθειες προτύπων για τις βασικές πύλες περιλαμβάνουν την ηχητική τηλεοπτική διαλειτουργικότητα (HAVi), τη συμμαχία τηλεφωνικής δικτύωσης (HPNA) και την ανοικτή πρωτοβουλία υπηρεσιών πυλών (OSGi).

1.4 Χρειαζόμαστε πραγματικά ένα Home Gateway;

Γενικά την δυνατότητα κάλυψης μιας βασικής πύλης μπορεί να ταιριαστεί με ένα σύνολο εναλλακτικών προϊόντων δικτύωσης. Παραδείγματος χάριν, ένα βασικό δίκτυο με ένα hub, τη σύνδεση της Microsoft Internet Connection Sharing (ICS), και ένα προσωπικό firewall όπως το ZoneAlarm περιέχει ήδη την ουσιαστική λειτουργία που ένα Residential Gateway παρέχει.

Ένα Residential Gateway θα κοστίζει γενικά περισσότεροι από την άποψη της up-front αγοράς από τις εναλλακτικές λύσεις, αλλά το κόστος αυτό, μπορεί να ανακτηθεί από την αποταμίευση στον χρόνο εγκαταστάσεων και συντήρησης. Εκείνοι που στερούνται το ενδιαφέρον

ή το ικανοποιητικό υπόβαθρο στην τεχνολογία δικτύωσης υπολογιστών θα βρουν πιθανώς την περισσότερη έκκληση στα Residential Gateway. Οι υπερασπιστές της τεχνολογίας ισχυρίζονται ότι τα προϊόντα τους προσφέρουν υψηλότερα επίπεδα αξιοπιστίας από τους κεντρικούς υπολογιστές PC στα client/server setups και περισσότερα χαρακτηριστικά γνωρίσματα από το συμβατικό Ethernet switch ή το δρομολογητή, όπως η υποστήριξη για τους ασύρματους χρήστες.

Το περιεχόμενο της εργασίας

2.0.1 MPC8260 Manual Study

Η συνεργασία μας με την Ellemedia Technologies ξεκίνησε στις αρχές του Μαΐου την χρονιά 2001. Λόγο της μικρής μας εμπειρίας στο Hardware, ξεκινήσαμε με την μελέτη του εγχειριδίου του επεξεργαστή MPC8260 της Motorola.

Μάθαμε όλα για την διαδικασία που έπρεπε να χρησιμοποιηθεί για το configuration του Motorola MPC8260 επεξεργαστή. Τί είναι το HRCW (Hard Reset Configuration word), και πώς χρησιμοποιείται. Διάβασαμε όλα για τους Memory map registers (ORx, BRx), τους registers ελέγχου της SDRAM (PSDMR), τον watchdog timer και το Baud rate generation system (BRGs).

Μάθαμε επίσης για τον τρόπο που μπορούμε να διαμορφώσουμε τα διάφορα interfaces. Πώς μπορούμε να οργανώσουμε τους Fast Communication Controllers (FCCs), και πώς μπορούμε οργανώσουμε την διασύνδεση ΥΤΟΡΙΑ II.

2.0.2 Boot Loader Study

Μιας και η πρόθεσή μας ήταν να κάνουμε import τον linux kernel στο δικό μας ιδιαίτερο MPC8260 board, έπρεπε να κάνοθμε πρώτα import ένα monitor πρόγραμμα, το οποίο θα αρχικοποιούσε τον επεξεργαστή και την πρωτότυπη πλακέτα.

Η Motorola δίνει ένα μικρό τέτοιο boot loader. Μελετήσαμε τον κώδικα, και καταλάβαμε πως μπορούμε να γράψουμε έναν μικρό αλλά ευέλικτο interrupt handler. Καταλάβαμε πως να ελέγχουμε και πώς να αρχικοποιούμε την σσιίδα μνήμης και πώς να αρχικοποιούμε την σειριακή κονσόλα στο Serial Communication Controller(SCC).

Επίσης είχαμε την ευκαιρία να μελετήσουμε την αρχιτεκτονική εντολών (ISA) του Motorola 603e core. Καταλάβαμε πώς εκτελούντε οι αριθμητικές πράξεις, πώς ελέγχουμε το σύστημα ύστερα από επιστροφή από διακοπή (rfi instruction), όπως επίσης και την εσωτερική δομή

του επεξεργαστή. Αυτό συμπεριλαμβάνει των αριθμό των γενικού σκοπού καταχωρητών, τον τρόπο που χρησιμοποιούντε οι καταχωρητές BAT για την μετάβαση από την φυσική στην ιδεατή μνήμη και την σειρά που χρησιμοποιούμε για να ενεργοποιήσουμε ή να απενεργοποιήσουμε τις Data, Instruction Caches.

2.0.3 Εισαγωγή του Boot Loader

Πρώτα έπρεπε να μάθουμε και να καταλάβουμε πως δούλευε ο OCD Commander της Macraigor Technologies. Αφού καταλάβαμε πώς λειτουργεί το πρόγραμμα αυτό, γράψαμε μικρά προγράμματα σε κώδικα μηχανής έτσι ώστε να έχουμε τη δυνατότητα να κάνουμε debug το InfoStar C1 board. Τα προγράμματα αυτά είχανε πολλές και διάφορες λειτουργίες. Για παράδειγμα ενεργοποιούσε και απενεργοποιούσε το ethernet PHY. Αυτό για να να μπορούμε να ελέγξουμε αν δουλεύει σωστά η ethernet (μιας και τα λαμπάκια του controller αναβοσβήνανε).

Στη συνέχεια το επόμενο βήμα ήταν η εισαγωγή του rrcboot boot loader. Εδώ είχαμε να κάνουμε πολλές ρυθμίσεις. Έτσι έπρεπε να καταλάβουμε σε βάθος όλη την διαδικασία αρχικοποίησης του επεξεργαστή και του περιβάλλοντος. Ύστερα από όλα αυτά, και ύστερα από έναν μήνα εργασίας, μπορέσαμε να τρέξουμε με επιτυχία το rrcboot και να δούμε το prompt στην σειριακή κονσόλα. Αλλά για τον προγραμματισμό του monitor προγράμματος στη μνήμη είχαμε μόνο το JTAG debugger. Έτσι έπρεπε να γραφτεί ένα πρόγραμμα σε C, που να μετατρέπει το SREC αρχείο του rrcboot σε ένα MAC αρχείο του OCD Commander. Αλλά μιας και το JTAG είναι αργό, χρειαζόμασταν περίπου μία και μιση ώρα για να προγραμματίσουμε μόνο μία φορά το monitor πρόγραμμα στη μνήμη.

Στη συνέχεια έπρεπε να ρυθμίσουμε το κομμάτι της ethernet του monitor προγράμματος. Για να επιτευχθεί ο σκοπός αυτός, έπρεπε να κατανοήσουμε τον τρόπο με τον οποίο λειτουργούσε η ethernet. Έτσι ύστερα από δύο μέρες εργασίας μπορούσαμε να κατεβάζουμε αρχεία στην μνήμη του συστήματος μέσω του πρωτοκόλλου TFTP.

Άλλο ένα σημαντικό εμπόδιο στην προσπάθειά μας, ήταν η μετατροπή του flash driver του rrcboot. Το rrcboot υποστήριζε μόνο 32 bit wide flash chips. Αλλά εμείς στο InfoStar C1 είχαμε 64 bit wide data bus. Έτσι αναγκαστήκαμε να αλλάξουμε σχεδόν όλο το αρχείο flash.c. Οι αλλαγές στον κώδικα ήταν περίπου 500 σειρές.

2.0.4 Linux Kernel Import

Αν και το rrcboot έτρεχε στην πλακέτα μας,ο linux kernel απαιτούσε πραγματικά σκληρή εργασία. Την πρώτη φορά που τον εισάγαμε δεν πήραμε κανένα αποτέλεσμα. Δεν μπορούσαμε να ξέρουμε που ο πυρήνας crashed, δεδομένου ότι δεν είχαμε καμία έξοδο στην

κονσόλα. Έτσι ήταν πραγματικά δύσκολο το debugging. Έπρεπε να γράψουμε κώδικα assembly, μέσα στο αρχείο head.S, στο οποίο γράφαμε ειδικούς αριθμούς (παραδείγματος χάριν 0xDEADC0DE) σε ειδικές θέσεις μνήμης. Με τον τρόπο αυτό μπορούσαμε να δούμε εάν ο μετρητής προγράμματος (PC) πέρασε από το ένα σημείο ελέγχου. Ξεδιάστηκα να αλλάξουμε τον πίνακα πληροφοριών (board info table), τον οποίο το prcboot πέρναγε στον πυρήνα, έτσι ώστε να έχουμε τα σωστά στοιχεία έναρξης.

Μετά από αυτό, πήραμε κάτι στην σειριακή κονσόλα μας, αλλά ήταν αδύνατο για μας να την διαβάσουμε. Έδειχνε λε και το baudrate δεν ήταν σωστό. Μετά από μερικές ημέρες debugging και μέσω της λίστας prc-embedded, λάβαμε ένα patch που έκανε τις κατάλληλες αλλαγές για την σειριακή κονσόλα να τρέχει στο SCC και όχι στον ελεγκτή SMC που έτρεχε αρχικά. Έτσι τελικά είχαμε τη σωστή σειριακή έξοδο. Μετά από λίγες ημέρες, ήμασταν σε θέση να έχουμε ένα πλήρως λειτουργικό Residential Gateway.

2.0.5 Η δημιουργία του Infostar C3

Στη συνέχεια έπρεπε να αφήσουμε το μέρος λογισμικού και να πάμε στο μέρος υλικού με το να σχεδιάσουμε ένα νέο προτότυπο σύστημα. Αυτό που πραγματικά έπρεπε να κάνουμε, ήταν να λάβουμε ένα έτοιμο σχέδιο ενός πρωτότυπου που είχε τρία pot interfaces, και να το αλλάξουμε έτσι ώστε να υποστήριζε τρία ISDN interfaces.

Το αρχικό σχέδιο είχε 2 γρήγορα ethernet interfaces, ένα ethernet οπτικό ινών, ένα interface PCMCIA, ένα interface USB, τις σειριακές θύρες, τον επεξεργαστή MPC8260 της Motorola και τρία pot interfaces. Έπρεπε να αφαιρέσουμε τα τρία pot interfaces, και να βάλουμε τρία ISDN interfaces. Χρειάστηκε να βάλουμε ένα μικρό FPGA στο σύστημα για να κάνουμε τη σωστή διασύνδεση.

Το σχηματικό δημιουργήθηκε χρησιμοποιώντας το λογισμικό Accel PCAD [1]. Η δρομολόγηση της πλακέτας έγινε χρησιμοποιώντας το λογισμικό Cadance Specctra [2]. Και τα δύο προγράμματα ήταν περίπλοκα και χρειάστηκε πολύ χρόνο για να τα μαθούμε.

Χρειάστηκε να σχεδιάσουμε το πρωτότυπο σύστημα δύο φορές. Ένα με μία FPGA στη συσκευασία BGA και ένα με μία FPGA στη συσκευασία TQFP. Τέλος, η δρομολόγηση ήταν πραγματικά δύσκολη, δεδομένου ότι έπρεπε να δρομολογήσουμε σχεδόν από την αρχή όλη την πλακέτα.

2.0.6 Motorola VADS development board

Αφού τελειώσαμε όλη την ανωτέρω εργασία, σειρά είχε το import του prcboot και του linux kernel στο board ανάπτυξης Motorola VADS. Αν και φαίνεται να είναι τετριμμένο, χρειάστηκε και να μάθουμε πολλές ειδικές λειτουργίες της πλακέτας έτσι ώστε να μπορούμε να ολοκλη-

ρώσουμε εργασία μας. Παραδείγματος χάριν χρειάστηκα να προσθέσουμε υποστήριξη για το BCSR ¹ στο prcboot και στον πυρήνα Linux.

Τέλος είχαμε να εκτελέσουμε μερικές τροποποιήσεις στον οδηγό της flash καθώς επίσης και στον οδηγό της ethernet. Οι τροποποιήσεις έγιναν περισσότερο στον prcboot boot loader.

2.0.7 Μελέτη του Motorola ATM Interface

Η δημιουργία ενός οδηγού για linux χρειαζόταν πάρα πολλή σκληρή εργασία. Όλα χρειάστηκαν να γίνουν με ακρίβεια. Ο οδηγός έπρεπε να ακολουθεί κατά γράμμα το εγχειρίδιο του MPC8260 επεξεργαστή. Έπρεπε να αρχικοποιεί όλες τις δομές και τους καταχωρητές στους οποίους αναφέρεται το εγχειρίδιο.

Έτσι το πρώτο πράγμα που έπρεπε να κάνουμε, ήταν να διαβάσουμε το εγχειρίδιο και να γίνει κατανοητή σε βάθος η λειτουργία του πυρήνα του ATM SAR. Στο διαδύκτιο, μπορείτε να βρείτε έναν κώδικα GPL, για τον οδηγό του ATM για τον επεξεργαστή 860. Έτσι έπρεπε επίσης να μελετήσουμε το εγχειρίδιο του Motorola MPC860, έτσι ώστε να μπορούμε να καταλάβουμε ποιες τροποποιήσεις έπρεπε εκτελέσουμε στον αρχικό κώδικα.

Ένα άλλο πράγμα που μας βοήθησε να καταλάβουμε τον τρόπο που αρχικοποιούμε τον πυρήνα του ATM SAR, ήταν ένα μικρό αυτόνομο πρόγραμμα που διένειμε η Motorola μαζί με τον επεξεργαστή. Το μόνο που έκανε το πρόγραμμα αυτό, ήταν να διαβιθαστεί ένα μικρό αριθμό AAL5 πακέτων και μέσω του PHY loopback να τους συλλέξει.

Αν και αυτό το πρόγραμμα ήταν αρκετά απλό, κατορθώσαμε να καταλάβουμε πολλά βασικά στοιχεία για το πώς δημιουργούμε έναν driver για linux kernel.

2.0.8 Η δημιουργία του ATM driver

Τελικά, μετά από τη μελέτη του εγχειριδίου, αλλάξαμε τον οδηγό του ATM του επεξεργαστή 860, έτσι ώστε να υποστηρίζει τον επεξεργαστή Motorola MPC8260. Οι αλλαγές που έπρεπε να εκτελέσουμε, ήταν περίπου 3000 γραμμές κώδικα. Νέες δομές έπρεπε να δημιουργηθούν, καθώς και νέα ακολουθία αρχικοποίησης.

Αλλά για δουλέψει σωστά ο driver με την δική μας πλακέτα, έπρεπε να γράψουμε έναν μικρό οδηγό για το IDT77v106 ATM PHY. Ο δίαυλος ελέγχου αυτού του PHY ήταν συνδεδεμένος με τους καταχωρητές GPIO, και έπρεπε να δημιουργήσουμε όλους τους απαιτημένους χρονοισμούς. Αυτός ο κώδικας ήταν περίπου 500 γραμμές.

¹BCSR είναι ένα μικρό σύνολο απο καταχωρητές, συγκεκριμένους για το αναπτυξιακό board

Το Residential Gateway

Η φράση Residential Gateway (RG) έγινε ένας καυτός όρος βιομηχανίας στα τέλη του 1995 με την έκδοση ενός άσπρου εγγράφου από μια άτυπη βιομηχανία η κοινοπραξία - Residential Gateway Group , ή η ομάδα RG ¹.

Στο άσπρο έγγραφό της, η ομάδα RG πρόβλεψε μια τυποποιημένη και ευφυή συσκευή διαπροσωπειών δικτύων που τερματίζει όλα τα εξωτερικά δίκτυα πρόσβασης (twisted pair, καλώδιο, δορυφόρος, κ.λπ....) και επιτρέπει ουσιαστικά οποιαδήποτε υπηρεσία σε οποιαδήποτε συσκευή μέσω των δικτύων εσωτερικής καλωδίωσης. Αν και αυτό το όραμα δεν έχει υλοποιηθεί ακόμα, το άσπρο έγγραφο της ομάδας RG έχει επηρεάσει έντονα τις επόμενες προσπάθειες ανάπτυξης Residential Gateway. Οι περισσότερες από αυτές τις προσπάθειες, εντούτοις, έχουν στραφεί στην ανάπτυξη ιδιόκτητων προϊόντων

Εν τω μεταξύ, αρκετές οργανώσεις - TIA (Telecommunications Industry Association) TR41.5 Committee, ISO/IEC (International Organization for Standardization/International Electrotechnical Commission), and The Open Services Gateway Initiative (OSGi) - έχουν προωθήσει χωριστές προσπάθειες που στοχεύουν στη δημιουργία τεχνικών προδιαγραφών για τα Residential Gateway.

3.1 Επισκόπηση των Residential Gateway

Μια πύλη συνδέει τις ανόμοιες αρχιτεκτονικές και παρέχουν end-to-end συνδετικότητα πέρα από το δίκτυο. Το Residential Gateway εκτελεί αυτήν την λειτουργία από τη σύνδεση των εξωτερικών (πρόσβαση) και εσωτερικών (in-home) δικτύων για το σπίτι.

Ένα Residential Gateway έχει τρία βασικά χαρακτηριστικά. Πρώτα, είναι PC-independent.

¹Το RG group αποτελούταν από αντιπροσώπους από την David Sarnoff Research Center, Hewlett-Packard Co., Bell Communications Research Inc. (or Bellcore, τώρα Telcordia Technologies), BnC Consulting, GTE Corp. (τώρα Verizon), IBM Corp., και RELTEC Corp. Αυτή η ομάδα δεν είναι πλέον ενεργή.

Δεύτερον, είναι ικανό να ρυθμίζει την κυκλοφορία μεταξύ των WAN και πολλαπλάσιων συσκευών στο σπίτι. Τρίτον, είναι μια enabling platform για τις υπηρεσίες/εφαρμογές. Τα περισσότερα από τα προϊόντα πυλών που είναι αναπτυγμένα μέχρι σήμερα βρίσκονται κάτω από τις ακόλουθες δύο κατηγορίες:

- (1). **Broadband Gateways.** Αυτές οι πύλες έχουν ένα ψηφιακό modem, λειτουργία δρομολόγησης, και ένα ή περισσότερα home networking interfaces built in. Μερικοί από αυτούς ενσωματώνουν επίσης πολλαπλά voice ports και μπορούν έτσι να ταξινομηθούν ως "multi-service broadband gateways". Ανάλογα με τον αριθμό των LAN ports, ή τύπων interfaces του τοπικού LAN, μια broadband πύλη μπορεί να απαιτήσει τη χρήση μιας χωριστού hub δικτύωσης ή ενός μεταλλάκτη μέσων για τη συνδετικότητα του τοπικού LAN.
- (2). **Stand-alone LAN Routers.** Αυτές οι συσκευές διασυνδέονται με ένα broadband modem (DSL, καλώδιο, ή wireless) και επιτρέπει τη συνδετικότητα του τοπικού LAN μέσω Ethernet ή κάποιου άλλου τύπου interface βασικής δικτύωσης (π.χ., οι δρομολογητές του τοπικού LAN, HPNA ή wireless). Οι LAN δρομολογητές, σχεδιάζονται κανονικά μόνο για την κοινή πρόσβαση στο Διαδίκτυο και μπορούν να θεωρηθούν λογικές πύλες όταν συνδυάζονται με ένα broadband modem.

Άλλα προϊόντα RG που είναι έτοιμα ή είναι ακόμα υπό ανάπτυξη [6], περιλαμβάνουν την telemetry/thin server gateways (πρώτιστα που επιτρέπουν την απομακρυσμένη βασική πρόσβαση, έλεγχο, και διαχείριση), και τις and set-top gateways (συνδυασμός ενός set-top box, ενός broadband modem, και μιας πλήρους βασικής δικτύωσης), και τα whole-house gateways (που προβλέπονται από την ομάδα RG).

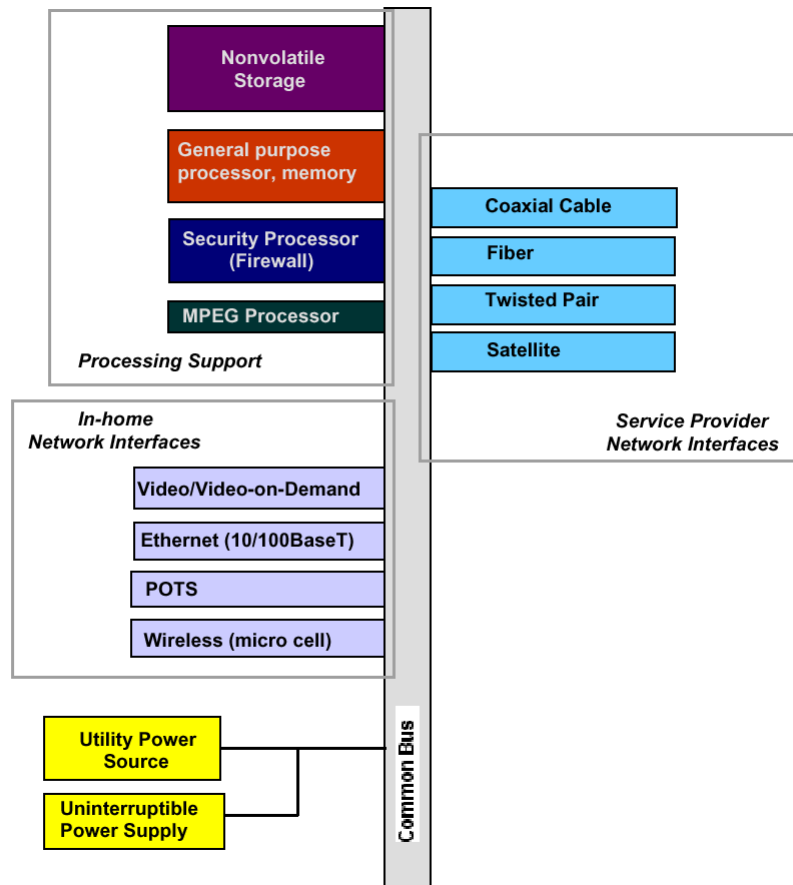
Οι αναλυτές βιομηχανίας και η βιομηχανία RG έχουν βρει πολλαπλάσιους τρόπους να ταξινομήσουν τα Residential Gateway.

3.2 Η Αρχιτεκτονική των Residential Gateway

Η αρχιτεκτονική των Residential Gateway προτάθηκε από την αρχική ομάδα RG στο άσπρο έγγραφο [4] και στα λειτουργικά specifications [5]. Η αρχιτεκτονική αυτή παρουσιάζεται στο 3.1.

Υπάρχουν πέντε λειτουργικά συστατικά της residential gateway: ένας κοινός δίαυλος, ανεφοδιασμοί ισχύος, διαπροσωπείες-βασικών δικτύων, δίκτυο υπηρεσιών προμηθευτών, και επεξεργασία υποστήριξης. Αυτή η αρχιτεκτονική παρέχει ένα κατάλληλο τμήμα μεταξύ του προμηθευτή υπηρεσιών και των καταναλωτικών συστατικών, που παρέχουν ένα σαφές τμήμα της ευθύνης κόστους και συντήρησης.

Σχήμα 3.1: RG Architecture



Ο κοινός δίαυλος εξυπηρετεί ως κοινή διαπροσωπεία υλικού μεταξύ των δικτύων προμηθευτών υπηρεσιών και του οικιακού δικτύου. Παρέχει την κοινή φυσική σύνδεση μεταξύ των συστατικών πυλών. Αυτά τα συστατικά θα είναι πιθανώς η ευθύνη του καταναλωτή. Οι ανεφοδιασμοί ισχύος παρέχουν την ισχύ στην πύλη για την κανονική λειτουργία. Αυτή η έννοια πυλών απαιτεί έναν uninterruptible ανεφοδιασμό ισχύος (UPS) για να διατηρήσει την υποστήριξη συνεχής επικοινωνίας κατά τη διάρκεια των δημόσιων διακοπών στην παροχή ηλεκτρισμού. Όπως με τον δίαυλο, αυτό το συστατικό θα είναι πιθανώς η ευθύνη του καταναλωτή.

Οι διαπροσωπείες-βασικών δικτύων παρέχουν την αλληλοσύνδεση μεταξύ των υπηρεσιών πληροφοριών που διανέμονται στην πύλη (προγραμματισμός καλωδιακής τηλεόρασης, φωνή, κ.λ.π.) και οι καταναλωτικές συσκευές που χρησιμοποιούν αυτές τις υπηρεσίες (τηλεόραση, τηλέφωνο, κ.λπ.). Τα εισερχόμενα σήματα στον δίαυλο θα αποκτιόνταν από την κατάλληλη διαπροσωπεία για την διανομή στο κατάλληλο δίκτυο ή τη συσκευή πληροφοριών [3]. Οι συγκεκριμένες διαπροσωπείες θα απαιτούνταν μόνο για τις υπηρεσίες δικτύων που επεκτείνονται

νονται στο σπίτι. Οι διαπροσωπείες-βασικών δικτύων θα ήταν η ευθύνη του καταναλωτή.

Οι διαπροσωπείες δικτύων προμηθευτών υπηρεσιών εκτελούν μια παρόμοια λειτουργία καθώς το βασικό δίκτυο interfaces. Παρέχουν την αλληλοσύνδεση μεταξύ του δικτύου προμηθευτών υπηρεσιών και της κατοικημένης πύλης. Οι διαπροσωπείες θα απαιτούνταν μόνο για τις υπηρεσίες που αγοράζονται από τον καταναλωτή και θα ήταν η ευθύνη του προμηθευτή υπηρεσιών. Πρέπει να είναι προσιτοί από έξω από την κατοικία.

Οι γενικοί και ειδικοί σκοπού επεξεργαστές παρέχουν τις λειτουργίες υποστήριξης επεξεργασίας. Αυτές οι λειτουργίες μπορούν να περιλάβουν την πιστοποίηση ταυτότητας, την καθοδήγηση και κωδικοποίηση/αποκωδικοποίηση του υλικού. Η υποστήριξη επεξεργασίας θα ήταν η ευθύνη του καταναλωτή αλλά μπορεί να υποστηριχθεί από το ειδικό λογισμικό που παρέχεται από τους προμηθευτές υπηρεσιών.

3.3 The Open Service Gateway Initiative

Ένα άλλο αναδυόμενο κατοικημένο μοντέλο πυλών προέρχεται από τη "ανοικτή πρωτοβουλία πυλών υπηρεσιών" (OSGi). Αυτή η πρωτοβουλία υποστηρίζεται από μια κοινοπραξία των βασικών φορέων βιομηχανίας όπως η Ericsson, της IBM, του NCI, Nortel, Motorola και άλλα, οι οποίοι ελπίζουν να αναπτύξουν οικιακά πρότυπα πυλών που η βιομηχανία θα υιοθετήσει. Η έννοιά τους είναι βασισμένη στο θεμέλιο οικοδομής ότι "το δικτυωμένο σπίτι είναι το επόμενο σύνδρομο, το Διαδίκτυο και οι νέες τεχνολογίες επιτρέπουν στις νέες υπηρεσίες, και ότι τα πρότυπα απαιτούνται για την αγορά για" να απογειωθούν", και ότι οι καταναλωτές θα απαιτήσουν μια end-to-end λύση που απαιτεί τη συνδετικότητα Διαδικτύου, για το βασικό εξοπλισμό [10].

Η προδιαγραφή OSGi είναι ένα Java-based application layer framework (το ISO/IEC Residential Gateway Model είναι επίσης ένα application layer framework που δίνει τους προμηθευτές υπηρεσίας, τους χειριστές δικτύων, τον προμηθευτή-ουδέτερο εφαρμογή κατασκευαστών συσκευών και κατασκευαστών συσκευών και το APIs layer συσκευών και τις λειτουργίες. Από το σχέδιο, η προδιαγραφή OSGi πρέπει να είναι ανεξάρτητος πλατφορμών έτσι μπορεί να εφαρμοστεί σε ποικίλους υπολογισμούς, επικοινωνίες, και πλατφόρμες και προϊόντα συσκευών. Χρειάζεται να είναι ανοικτά τα πρότυπα, μιας και αυτό θα παρουσιάσει τους προμηθευτές και τους προμηθευτές υπηρεσίας με μια κοινή διαπροσωπεία, και θα έχουμε έναν μακρύ δρόμο προς την παρεμπόδιση του ενιαίου ελέγχου επιχείρησης της αρχιτεκτονικής [7].

Το OSGi είναι μια συλλογή από APIs που καθορίζουν τα ανοικτά πρότυπα για μια οικιακή πύλη. Αποτελούνται από ένα σύνολο core και των προαιρετικών APIs που καθορίζουν τη OSG συμβατή πύλη. Η OSG προβάλλει τα πρότυπα της Java όπου είναι δυνατόν. Ο πυρήνας των APIs σχεδιάζεται για να εξετάσει την παράδοση υπηρεσιών, την εξάρτηση και τη διαχεί-

ριση κύκλων ζωής, τη διαχείριση στοιχείου συμπεριφοράς, την απομακρυσμένη διαχείριση υπηρεσιών, την απομακρυσμένη διοικητική μέριμνα και τη διαχείριση [10] υπηρεσιών συσκευών. Το προαιρετικό σύνολο των APIs θα καθορίσει πώς οι χρήστες αλληλεπιδρούν με την πύλη και πώς το θα ρυθμίζεται το στοιχείο. Θα περιλάβει αρκετών υπάρχουσα Java APIs (συμπεριλαμβανομένου και του JINI). Ένας προμηθευτής (κατάλληλα επικυρωμένος) που χτίζει OSG που η συμβατή πύλη πρέπει να εφαρμόσει τον πυρήνα των APIs, αλλά δεν απαιτείται για να περιλάβει όλα τα προαιρετικά APIs. Η OSG κοινοπραξία προδιαγραφών χρησιμοποιεί μια ανοικτή διαδικασία προτύπων που βασίζεται στην κοινοτική διαδικασία της Java.

3.4 Προτερήματα

Μιας που η πολυπλοκότητα της διαχείρισης των πολλαπλάσιων εξωτερικών δικτύων μέσα στις αυξήσεις κατοικιών, το RG μπορεί να παρέχει μια απλουστευμένη διαπροσωπεία που ενσωματώνει τις πολλαπλάσιες υπηρεσίες πρόσβασης με το-βασικό δίκτυο. Οι βελτιώσεις υπηρεσίας θα ήταν τόσο εύκολες όπως συνδέοντας μια κάρτα με μια αυλάκωση. Με το ενσωματωμένο διαμορφώσιμο θα είναι ευκολότερο για τους καταναλωτές να προσθέσουν τις νέες υπηρεσίες και να αφαιρέσουν εκείνες που δεν είναι πλέον χρήσιμες.

Στην ευκολία στην αλλαγή των υπηρεσιών προμηθευτών ωφελείται ένα άλλο καταναλωτικό όφελος. Εάν η υπηρεσία πελατών επιδεινώνει με τον έναν προμηθευτή, το RG επιτρέπει την εύκολη πρόσβαση σε έναν άλλο προμηθευτή. Αυτό αυξάνει τον ανταγωνισμό μεταξύ των προμηθευτών υπηρεσιών και των κατασκευαστών εξοπλισμού και οδηγεί στην παροχή των νέων υπηρεσιών και των χαμηλότερων τιμών στον καταναλωτή.

Το RG πρέπει να χαμηλώσει το καταναλωτικό κόστος από την εξάλειψη της ανάγκης για τους πολλαπλάσιους μετασχηματιστές και τους διαποδιαμορφωτές. Ένα προσωρινό βήμα θα μπορούσε να μετατοπίσει μερικές από τις λειτουργίες των μετασχηματιστών στο RG. Με την κίνηση της νοημοσύνης στο RG, οι μετασχηματιστές θα μπορούσαν να παραχθούν πιο φτηνά. Οι προμηθευτές υπηρεσιών ωφελούνται επίσης από την εφαρμογή της οικιακής πύλης. Μια πρότυπη διαπροσωπεία παρέχει ένα κοινό σύνολο πρωτοκόλλων που μπορεί να μειώσει το κόστος ανάπτυξης. Η διαμορφωσιμότητα επιτρέπει τις ευκολότερες βελτιώσεις στο σύστημα θέρμανσης, αερισμού και κλιματισμού. Στις περισσότερες περιπτώσεις αυτό σημαίνει ότι ο καταναλωτής εγκαθιστά τη νέα υπηρεσία με τέτοιο τρόπο που μειώνει το κόστος επέκτασης στους προμηθευτές υπηρεσιών. Το RG υποστηρίζει τα απομακρυσμένους διαγνωστικά και τον έλεγχο δικτύων, μειώνοντας το κόστος εργασίας που συνδέεται με τη διαχείριση δικτύων.

Ένα από τα βασικά οφέλη στους προμηθευτές υπηρεσιών προέρχεται από τη δυνατότητα να προσφερθούν οι νέες υπηρεσίες στην καταναλωτική αγορά. Δεδομένου ότι το RG επιτρέπει την πρόσβαση σε όλα οικιακά δίκτυα, θα είναι ευκολότερο να προσφερθεί μια συσσωρευμέ-

νη συσκευασία των υπηρεσιών όπως η φωνή, τα στοιχεία και το βίντεο. Αυτό παρέχει την πρόσβαση στις νέες αγορές που παράγουν τα νέα ρεύματα εισοδήματος. Οι κατασκευαστές εξοπλισμού είναι ο τρίτος δικαιούχος της οικιακής πύλης. Μπορούν να χαμηλώσουν το κόστος τους από το σχεδιασμό σε μια πρότυπη διαπροσωπεία. Από την ανάπτυξη των καρτών για τη μια πλατφόρμα, οι οικονομίες κλίμακας θα μειώσουν το κόστος.

Το RG διευκολύνει επίσης την καινοτομία. Με την εξάλειψη της ανάγκης να ξανασχεδιάσουν τα χαμηλής αξίας συστατικά, οι κατασκευαστές μπορούν να συγκεντρωθούν σε νέα χαρακτηριστικά γνωρίσματα και στη λειτουργία. Αυτό οδηγεί στις αυξανόμενες πηγές εισοδήματος δεδομένου ότι η καινοτομία υποκινεί τη διαφοροποίηση προϊόντων και το ανταγωνιστικό πλεονέκτημα.

Οι κατοικημένες πύλες θα μπορούσαν να ανοίξουν το μονοπάτι για τη μεγαλύτερη ευελιξία για τους καταναλωτές, τους προμηθευτές υπηρεσίας και τους κατασκευαστές εξοπλισμού. Δεδομένου ότι η βασική έννοια της οικιακής πύλης επιτρέπει οποιαδήποτε πρόσβαση προμηθευτών υπηρεσίας σε οποιαδήποτε καλωδίωση μέσα στο σπίτι, οι διάφοροι προμηθευτές θα πρέπει να ανταγωνιστούν βασισμένοι στην ποιότητα και την τιμή. Συνεπώς, η οικιακή πύλη θα μπορούσε να αυξήσει τον ανταγωνισμό και να ενθαρρύνει τις νέες υπηρεσίες μειώνοντας τις τιμές στον καταναλωτή.

3.5 Προκλήσεις

Από την καταναλωτική προοπτική ένα από τα μεγαλύτερα εμπόδια στην εφαρμογή της κατοικημένης πύλης είναι το κόστος, συμπεριλαμβανομένων των δαπανών RG, εγκαταστάσεων και συντήρησης. Είναι επίσης ασαφές πώς το RG θα εγκατασταθεί αρχικά στο σπίτι ενός καταναλωτή. Συνεπώς, ποιος αντέχει το κόστος εγκαταστάσεων δεν έχει ακόμα καθοριστεί.

Το RG μπορεί να φανεί όπως μια σύνθετη συσκευή που διαχειρίζεται μέσα στο σπίτι. Αν και η έννοια RG είναι βασισμένη σε μια άνευ ραφής ολοκλήρωση των δικτύων πρόσβασης με το βασικό δίκτυο, οι λεπτομέρειες ως προς πώς αυτό θα ολοκληρωνόταν αναπτύσσονται ακόμα. Αυτή η διαφάνεια θα είναι κρίσιμη στο κέρδος της καταναλωτικής αποδοχής του RG.

Ένα άλλο σημαντικό εμπόδιο είναι η έλλειψη καταναλωτικής συνειδητοποίησης της οικιακής έννοιας πυλών [9]. Χρειάζεται φάση "ώθησης τεχνολογίας" και όχι φάση τραβήγματος αγοράς. Για να εμπορευτεί επιτυχώς αυτήν την έννοια θα απαιτήσει την καταναλωτική εκπαίδευση στα οφέλη και οι ανάγκες για προμηθευτές μιας RG. Οι μεγάλοι service providers θα μπορούσαν να είναι το σημαντικό οδόφραγμα για την εφαρμογή του RG. Αν και μπορούν να παραγάγουν πολλά οφέλη, χάνουν τον έλεγχο πρόσβασης στο σπίτι. Όπως οι τοπικοί μεταφορείς ανταλλαγής (LEC) έχουν ήδη καταδείξει, είναι πολύ απρόθυμοι να ανοίξουν τα δίκτυά τους στους πιθανούς ανταγωνιστές ακόμα κι αν η πράξη τηλεπικοινωνιών του 1996 εξουσιοδοτεί αυτό. Όταν η απώλεια ελέγχου πρόσβασης συνδέεται με τον αυξανόμενο αντα-

γωνισμό για όλες τις υπηρεσίες στο σπίτι, μπορεί να είναι δύσκολο να μαζευτεί η υποστήριξη που απαιτείται για την εφαρμογή του RG.

Ένα άλλο μειονέκτημα του RG για τους προμηθευτές υπηρεσίας είναι η δυνατότητά του να αυξήσει το σύνολο πελατών με το να καταστήσει εύκολη την ελαγή των προμηθευτές. Οι μεγάλης απόστασης και κυψελοειδείς τηλεφωνικοί προμηθευτές είναι εξοικειωμένοι με αυτό το ζήτημα. LECs και οι επιχειρήσεις καλωδίων δεν είχαν την ανάγκη να εξετάσουν το σύνολο πελατών δεδομένου του ότι έχουν ακόμα τα εικονικά μονοπώλια για τις τελευταίες συνδέσεις μιλίου.

3.6 Υπάρχοντα Residential Gateways

Η Ellemedia Technologies έχουν σχεδιάσει και έχουν αναπτύξει τέσσερα διαφορετικά οικιακά πρωτότυπα πυλών. Τα κώδικα ονόματα για αυτά τα πρωτότυπα είναι Infostar C1, Infostar C2, Infostar C3 και Infostar EP.

Όλα αυτά τα πρωτότυπα έχουν την ίδια βάση. Το πρώτο κατοικημένο πρωτότυπο που δημιούργησε η Ellemmedia Technologies, ήταν το InfoStar. Όλα τα νέα πρωτότυπα είναι μια απλούστερη έκδοση του αρχικού πρωτοτύπου.

Το InfoStar, είχε μερικά πρωτοπορίες. Το πρωτότυπο, αποτελούσαν από μια μεγάλη μητρική κάρτα και μερικές κάρτες επέκτασης. Η μητρική κάρτα είχε έναν επεξεργαστή ARM, τη μνήμη SDRAM και FLASH, καθώς επίσης και ένα υτοπία backbone. Σε αυτό το δίκτυο, θα μπορούσαμε να παρεμβάλουμε τις κάρτες επέκτασης προκειμένου να δημιουργηθεί η οικιακή πύλη της επιλογής μας. Μερικές από τις κάρτες επέκτασης είχαν υποστήριξη ATM, υποστήριξη τοπικού WaveLAN, ή ακόμα και την υποστήριξη τηλεφωνίας.

Με το σύνολο αυτών των καρτών θα μπορούσαμε να δημιουργήσουμε μία RG της επιλογής μας. Το πρόβλημα με αυτήν την προαιρετική δυνατότητα, ήταν η τιμή. Όμως, μιας και τα RGs, είναι για την οικιακή χρήση, μπορούμε να καταλάβουμε ότι θα προτιμούσαμε το χαμηλότερη δυνατή τιμή. Για το λόγο αυτό, η Ellemmedia Technologies δημιούργησε τις μικρότερες και φτηνότερες οικιακές πύλες της επιλογής μας.

InfoStar C3

4.1 Εισαγωγή

Το InfoStar/C3 είναι μία λύση οικιακής πύλης ανεπτυγμένης, χαμηλού κόστους, ISDN πρόσβασης για το σπίτι. Είναι μια ιδανική συσκευή πρόσβασης για το σπίτι, ένας συγκεντρωτής οικιακών δικτύων και ένας ελεγχτής ψυχαγωγίας όταν το μέσο φυσικής πρόσβασης είναι twisted pair. Προσφέρει οποιοδήποτε συνδυασμό υψηλής ταχύτητας δεδομένων, πρόσβαση στο διαδίκτυο και ψηφιακό video, γρήγορα και οικονομικά πάνω από ήδη υπάρχουσες γραμμές τηλεφώνου. Προσφέρει ενσωματωμένα, υψηλή ταχύτητα, πλήρη πρόσβαση ISDN, δύο αφοσιωμένα τμήματα LAN για το σπίτι διαμέσου δύο δρομολογητών 10/100 BaseT (RJ-45) Ethernet, μια κάρτα τύπου PCMCIA Type III συμβατή με τα καθιερωμένα standard, όπου οποιαδήποτε PC κάρτα μπορεί να ενσωματωθεί, και μια διασύνδεση για οπτική Ethernet.

Μέλος της οικογένειας προϊόντων InfoStar Compact, το InfoStar/C3 είναι σχεδιασμένο με τη φιλοσοφία της ελαχιστοποιημένης διαχείρισης και πραγματικής δυνατότητας plug-n-play. Στο κεντρικό σύστημα, τρέχει λειτουργικό Linux, προσφέρει αρχιτεκτονική ανοικτού λογισμικού και μια μεγάλη ποικιλία από drivers, πρωτόκολλα, και services. Web server, DNS, DHCP, NAT και Firewall είναι μία μικρή λίστα από δυνατότητες, ενώ το σύστημα είναι ανοιχτό σε οποιοδήποτε ώριμου ή προχωρημένου πρωτοκόλου, service ή εφαρμογή.

Με την ελαστικότητα να έχουμε οποιαδήποτε νέο interface στην υποδοχή κάρτας PCMCIA, το InfoStar/C3 στοχεύει σε νέες απαιτητικές αγορές για το σπίτι και για περιβάλλοντα SOHO, με μέσο φυσικής πρόσβασης το το καλώδιο twisted pair.

Features

- Compact/Flexible Design
- Embedded Microprocessor

- Full ISDN Access
- Up to 8Mbps Downstream
- Up to 1 Mbps Upstream
- 2 x 10/100 BaseT (RJ-45)
- 3 x ISDN interfaces
- PC Card (PCMCIA) Type III interface.
- Wireless LAN@11Mbps via Lucent's WaveLAN/Orinoco PCMCIA card.
- Advanced security via firewall and IPSec

Software Features

- Linux OS (Kernel 2.4)
- Flashdisk
- TCP/IP (RFC 791)
- PPP (RFC 1661)
- PPPoE
- PPP over ATM (RFC 1932/RFC 2364)
- Routed IP over ATM (RFC 1483)
- Classical IP over ATM (RFC 1577)
- ATM Signaling (Q2931)
- Q.SIG
- RSVP
- DiffServ
- IPSec
- Telnet
- Mini Web server

- DHCP server (RFC 1541)
- NAT server (RFC 1631)
- DNS server
- Firewall
- Remote access / management
- SNMP v2, v3 (RFC 1157, RFC 1458)

4.2 Αρχιτεκτονική

Όπως μπορούμε να δούμε στο σχήμα 4.1, η καρδιά όλου του συστήματος, είναι ο ενσωματωμένος επεξεργαστής Motorola MPC8260. Το σύστημα έχει 32MB μνήμη flash, και 128MB δυναμικής μνήμης. Εσωτερικά στην ανεξίτηλη μνήμη, είναι αποθηκευμένο ολόκληρο το λογισμικό. Αυτό συμπεριλαμβάνει το ppcboot boot loader, το λειτουργικό σύστημα Linux και το τοπικό filesystem. Και οι δύο αυτές μνήμες, είναι συνδεδεμένες με τον διαυλο 60x του επεξεργαστή. Αυτό προσφέρει γρήγορη πρόσβαση στην μνήμη, όπως επίσης και υποστήριξη για μεγάλες σε μέγεθος μνήμες. Και οι δύο αυτές μνήμες έχουν διαυλο 64bit wide.

Όλες οι κάρτες δικτύου, επικοινωνούν με τον επεξεργαστή διαμέσου του πρωτοκόλου MII (Media Independent Interface). Ο επεξεργαστής MPC8260, προσφέρει τρία port που υποστηρίζουν το παραπάνω πρωτόκολο επικοινωνίας. Για να μπορέσουμε να χρησιμοποιήσουμε τις τρεις κάρτες δικτύου, χρησιμοποιούμε τους τρεις ελεγχτές Fast Communication Controllers, (FCC1, FCC2, FCC3).

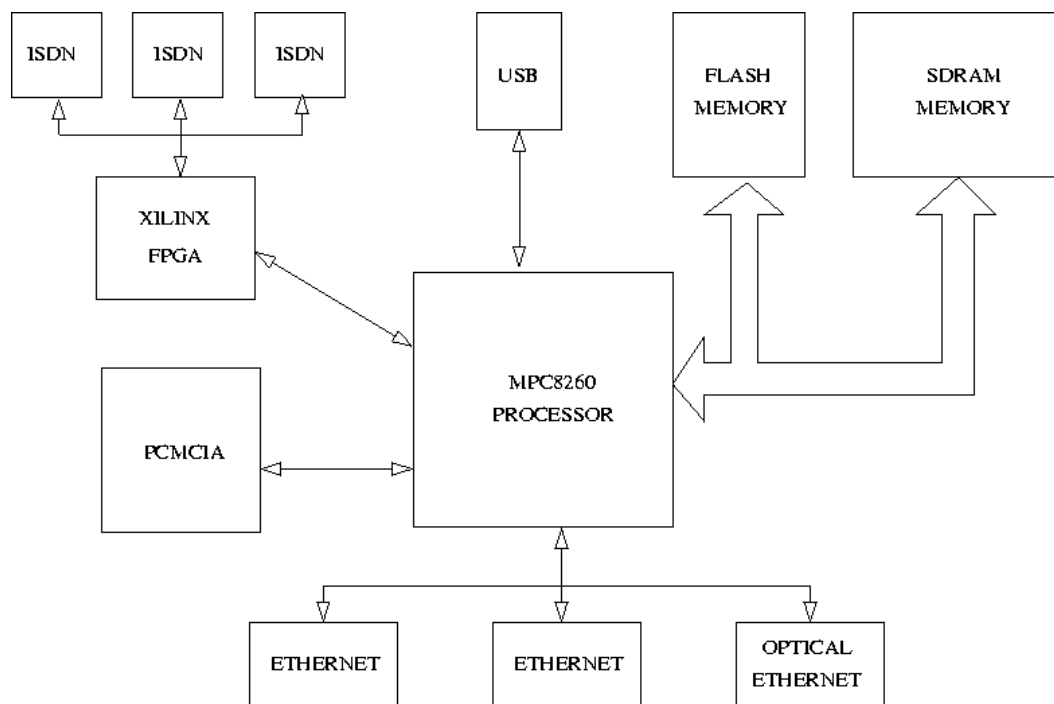
Η διασύνδεση PCMCIA είναι συνδεδεμένη στο τοπικό δίαυλο. Αυτό γιατί ο επεξεργαστής υποστηρίζει την διασύνδεση PCI. Το chipset που χρησιμοποιήθηκε, δεν είναι τίποτα άλλο από μία PCMCIA to PCI bridge.

Όλες οι άλλες διασυνδέσεις είναι συνδεδεμένες σε μία Xilinx FPGA. Τα τρία ολοκληρωμένα για ISDN, καθώς και η διασύνδεση USB, είναι συνδεδεμένα στα ελεύθερα I/O της Xilinx FPGA. Η Xilinx FPGA είναι συνδεδεμένη στον τοπικό δίαυλο του συστήματος, έτσι ώστε να έχουμε τη δυνατότητα να την προσπελάσουμε σαν μία εξωτερική μνήμη.

4.3 Η δημιουργία του σχηματικού

Το πρώτο που κάναμε ύστερα από το βασικό block διάγραμμα, ήταν να βρούμε το κατάλληλο σύνολο ολοκληρωμένων για την διασύνδεση της ISDN. Έπρεπε να λάβουμε υπ'όψη την ομαλή λειτουργία των ολοκληρωμένων με τον επεξεργαστή MPC8260 της Motorola. Έτσι

Σχήμα 4.1: InfoStar C3 Block Diagram



αναγκαστήκαμε να ψάξουμε τα βρούμε τα αντίστοιχα ολοκληρωμένα από διάφορες εταιρίας. Μερικές από αυτές είναι η nlineon, IDT, Lucent (Agere) και η Motorola. Η επιλογή δεν ήταν εύκολη διαδικασία. Παντού είχαμε θετικά και αρνητικά. Για παράδειγμα η Infineon πρόσφερε ένα πολύ φτηνό ολοκληρωμένο, αλλά είχε μια πολύ πολύπλοκη διασύνδεση που δεν ήταν εύκολο να υλοποιήσουμε.

Έτσι φτάσαμε στο συμπέρασμα πως η καλύτερη επιλογή ήταν το ολοκληρωμένο Motorola MC145574. Προσφερε μια πολύ ευκολη σε υλοποίηση διασύνδεση, και πάνω από όλα την βεβαιότητα ότι ο transceiver θα ήταν συμβατός με τον επεξεργαστή, μιας και ήταν κατασκευασμένος από την ίδια εταιρία.

Στη συνέχεια, έπρεπε να βρεθεί ο τρόπος με τον οποίο θα συνδεόντουσαν τα ολοκληρωμένα με τον επεξεργαστή. Η πιο εύκολη λύση ήταν διαμέσου της διασύνδεσης TDM που πρόσφερε ο επεξεργαστής. Όπως μπορούμε να δούμε απο το κεφάλαιο 14 του επεξεργαστή, χρειαζόμαστε τρία port για να κάνουμε τη διασύνδεση. Χρειαζόμαστε δύο SCC (Serial Communication Controller) και ένα SMC (Serial Managment Controller). Αλλα αυτό δεν ήταν εφικτό, μιας και χρησιμοποιούσαμε ήδη πολλές πόρτες για τα άλλα περιφεριακά. Μία απο τις επιλογές μας, ήταν να αφαιρέσουμε μία από τις περιφεριακές συσκευές. Όμως αυτό δεν ήταν εφικτό μιας και όλες ήταν απαραίτητες.

Έτσι η μόνη εναλλακτική λύση, ήταν να χρησιμοποιήσουμε μία χαμηλού κόστους FPGA

(Field Programmable Gate Array). Η ιδέα ήταν να μην έχουμε πρόσβαση στα ολοκληρωμένα διαμέσου των port του επεξεργαστή, αλλά διαμέσω μιας memory map area. Αυτό σημαίνει πως θα χρησιμοποιούσαμε την fpga, σαν μια ακόμα μνήμη του συστήματος. Έτσι έπρεπε να συνδέσουμε στην FPGA ένα σήμα chip select, μερικά από τα high order address bits, το σήμα Write Enable, το ras output enable signal, και βεβαίως τα 32bits από τον δίαυλο των δεδομένων. Μεσικά ακόμα σήματα, ήταν τα σήματα για τις διακοπές και μερικά ακόμα chip select σήματα.

Έτσι και έγινε. Επιλέξαμε την Xilinx Spartan-II XC2S100 FPGA. Αυτή η FPGA ήταν αρκετά μεγάλη για τις ανάγκες μας, και μάλιστα αρκετά φτηνή. Η FPGA αυτή, είχε ακόμα ένα μεγάλο αριθμό από ελεύθερα I/O. Και χρειαζόμασταν πολλά I/O pins για τη σωστή διασύνδεση. Κατα την ολοκλήρωση του σχηματικού, μόνο 10 I/O, ήταν ελεύθερα.

Το πιο κρίσιμο κομμάτι του σχηματικού, ήταν το αναλογικό. Το αναλογικό κομμάτι συμπεριλαμβάνει όλα τα αντικείμενα που ήταν συδεδεμένα ανάμεσα στα MC145574 και στον τελικό κονέκτορα της γραμμής. Μιας και τα σήματα της γραμμής είναι αναλογικά, έχουν μια διαφορετική συμπεριφορά. Τα πάντα έπρεπε να γίνουν με μεγάλη προσοχή.

Το σχηματικό δημιουργήθηκε με το πρόγραμμα PCAD 2001 Schematic της ACCEL Technologies.

4.4 Η δημιουργία του τυπωμένου κυκλώματος

Το επόμενο βήμα μετά την ολοκλήρωση του σχηματικού, ήταν η δημιουργία του τυπομένου κυκλώματος. Πρώτα δημιουργήσαμε μία netlist διαμέσω του λογισμικού του σχηματικού. Μετά εισάγαμε το αρχείο αυτό στο λογισμικό για την δημιουργία τυπωμένων κυκλωμάτων. Αλλά μιας και θέλαμε να αλλάξουμε ένα ήδη υπάρχον σχέδιο, έπρεπε πρώτα να δημιουργήσουμε ένα αρχείο ECO. Για να δημιουργήσουμε το αρχείο αυτό χρειαζόμασταν τη netlist από το καινούργιο σχηματικό, καθώς επίσης και το netlist του παλαιού τυπομένου κυκλώματος. Χρησιμοποιήσαμε το πρόγραμμα NetList Compare το οποίο και δημιουργεί το αρχείο ECO¹ που ζητάμε.

Έτσι εισάγαμε το αρχείο ECO στο λογισμικό σχεδίασης τυπομένων. Με την εισαγωγή του αρχείου αυτού, το πρόγραμμα αυτόματα σβήνει τα παλαιά κομμάτια που δεν χρειάζονται και εισάγει τα καινούργια. Αλλά τα στοιχεία αυτά πρέπει να τοποθετηθούν στα κατάλληλα σημεία στην πλακέτα. Έτσι μετακινούμε με το χέρι και τοποθετούμε κάθε στοιχείο στο σημείο που πρέπει.

Το πιο δύσκολο μέρος, ήταν το σβήσιμο των παλαιών καλωδίων. Όταν εισάγαμε το αρχείο ECO, αυτόματα σβήνει τα παλαιά κομμάτια που έχουν σβηστεί από το σχηματικό. Όμως δεν

¹Το αρχείο ECO περιέχει τις αλλαγές που χρειάζονται να γίνουν στο τυπομένο

σβήνει και τα καλώδια που ήταν δρομολογημένα προς το αντικείμενο αυτό. Έτσι έπρεπε να διαλέξουμε κάθε ένα καλώδιο με το χέρι, και να το διαγράψουμε.

Η τοποθέτηση των αναλογικών κομματιών ήθελε πολύ μεγάλη προσοχή. Πολλά από τα αναλογικά κομμάτια, έπρεπε να τοποθετηθούν στο κατάλληλο σημείο με τις κατάλληλες αποστάσεις από τα υπόλοιπα αναλογικά ή ψηφιακά κομμάτια. Επίσης μερικά από τα αναλογικά καλώδια, έπρεπε να έχουν και συγκεκριμένο πλάτος. Τέλος, έπρεπε να τοποθετηθεί και να προστεθεί και ένα plane στο τυπωμένο, για το αναλογικό ground, το οποίο είναι συνδεδεμένο στη κοινή γή, διαμέσου δύο πηνίων.

Υστερα από το καθάρισμα και την τοποθέτηση, έπρεπε να ξεκινήσουμε την δρομολόγηση των καλωδίων. Έτσι εισάγαμε το αρχείο του σχηματικού στο λογισμικό για δρομολόγηση SPECCTRA. Η δημιουργία μερικών βασικών .DO ² αρχείων, ήταν απαραίτητη. Έπρεπε να ορίσουμε το πάχος των καλωδίων, τον τύπο των VIA, να ορίσουμε μερικά κρίσιμα καλώδια, την απόσταση ανάμεσα σε δύο καλώδια και αρκετούς άλλους περιορισμούς. Όλοι αυτοί οι περιορισμοί εισαχθήκαν μέσα σε ένα αρχείο DO.

Υστερα από τον ορισμό όλων αυτών των περιορισμών, το τελικό βήμα ήταν η δρομολόγηση. Κάτι το οποίο αποδείχτηκε αρκετά δύσκολο και επίπονο. Έαν τρέχαμε την εντολή **route 10**, η δρομολόγηση που δημιουργούσε το λογισμικό δεν ήταν σωστή. Για παράδειγμα ένα απλό καλώδιο, μπορούσε να κάνει τον κύκλο όλου του τυπομένου για να φτάσει στον προορισμό του.

Υστερα από αρκετές μέρες σκληρής δουλειάς, διαπιστώσαμε, πως για να γίνει σωστή δρομολόγηση, έπρεπε να γίνει με το χέρι. Έτσι και έγινε. Περίπου το 60% του νέου κομματιού του τυπομένου δρομολογήθηκε με το χέρι. Για να γίνει βέβαια σωστή δρομολόγηση, χρειαστήκαμε να αλλάξουμε αρκετές φορές το σχηματικό, για να αλλάξουμε την θέση ορισμένων καλωδίων στην FPGA. Για παράδειγμα, εάν χρειαζόμασταν να αφαιρέσουμε ένα X mark ³ στο τυπομένο, έπρεπε να τα αντιστρέψουμε, και για να το κάνουμε αυτό έπρεπε να αλλάξουμε το σχηματικό.

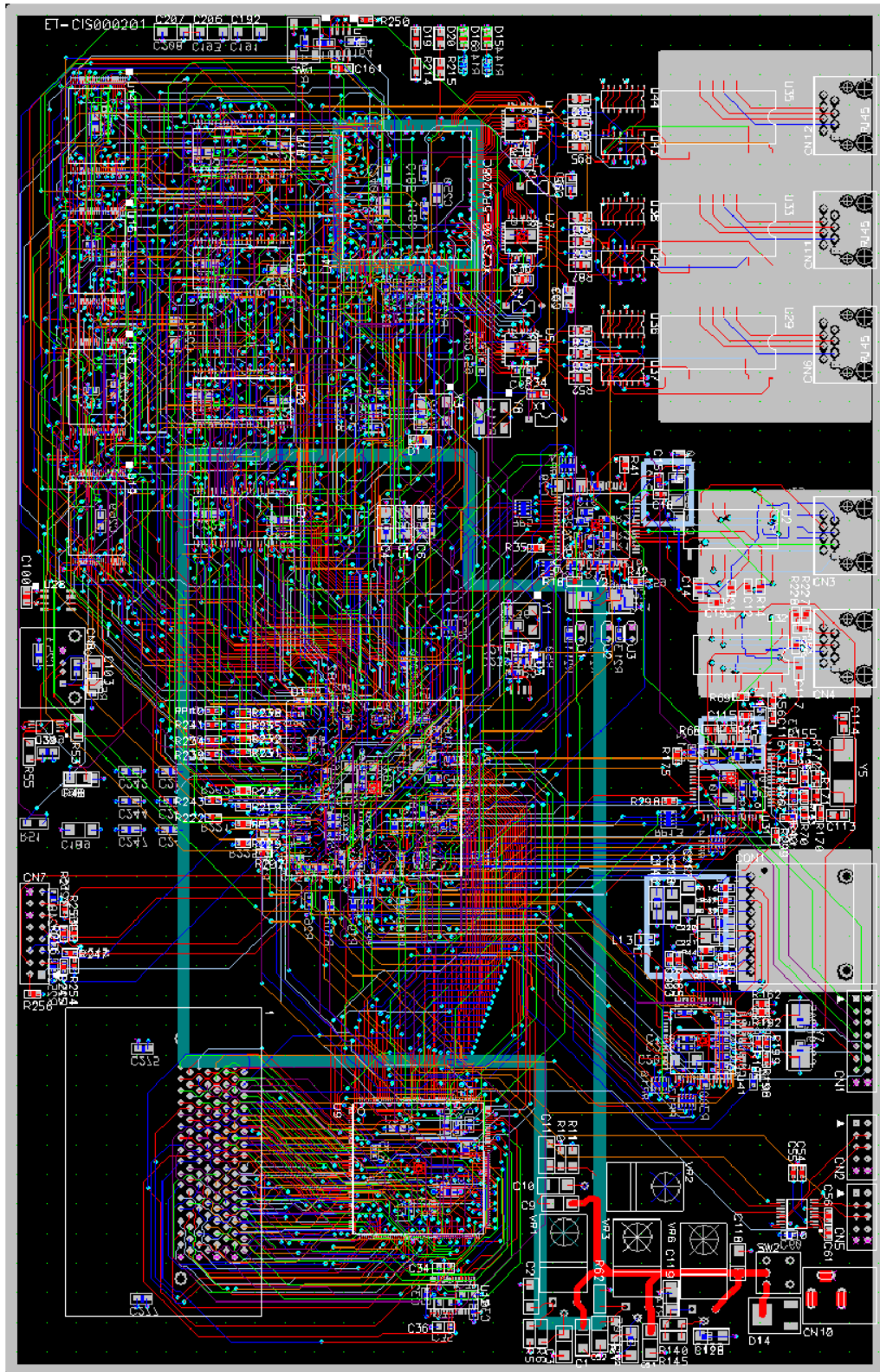
Αν αναλογιστούμε τον μεγάλο αριθμό των καλωδίων, μπορούμε να υπολογίσουμε τον χρόνο που χρειάζεται για να γίνει κάτι τέτοιο. Υστερα απο το τέλος του τυπομένου το τελικό βήμα ήταν η τοποθέτηση διαστασεων και μερικών άλλων πληροφοριών μέσα στο αρχείο του τυπομένου κυκλώματος.

Η δημιουργία του τυπομένου κυκλώματος έγινε με το λογισμικό PCB PCAD 2001 της ACCEL Technologies [1]. Η δρομολόγηση έγινε με το λογισμικό της Cadence, SPECCTRA v9.03, [2].

²Τα αρχεία .DO, είναι script files, που περιέχουν μερικές εντολές που θέλουμε να εκτελεστούν

³Το X mark ορίζουμε το σημείο που δύο καλώδια του ίδιου layer τέμνονται

Σχήμα 4.2: InfoStar C3 PCB



Motorola MPC8260 Processor

5.1 Ανασκόπηση της Αρχιτεκτονικής

Ο MPC8260 έχει δύο εξωτερικούς διαύλους για να ανταπεξέλθει στις απαιτήσεις του bandwidth από τον μεγάλης ταχύτητας επεξεργαστή και τα γρήγορα κανάλια Επικοινωνίας [8]. Όπως βλέπουμε στο σχήμα 5.1, ο MPC8260 έχει τρία βασικά λειτουργικά κομμάτια :

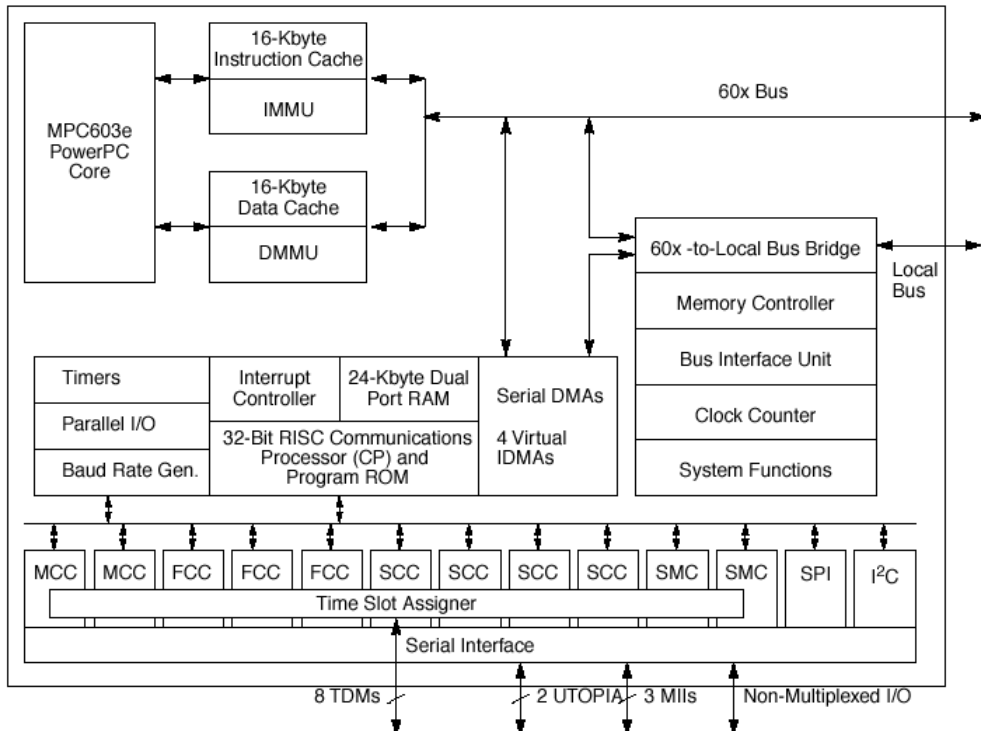
- Ένα 64-bit PowerPC core που πέρχεται από το MPC603e με MMUs και cache
- Ένα system interface unit (SIU)
- Ένα communications processor module (CPM)

Και ο πυρήνας του συστήματος και η CPM έχουν ένα εσωτερικό PLL, που επιτρέπει την ανεξάρτητη οπμιζατιον των συχνοτήτων στις οποίες τρέχουν. Ο πυρήνας του συστήματος και η CPM είναι και τα δύο συνδεδεμένα με το 60x δίαυλο.

5.2 MPC603e Core

Το σώμα του MPC603e έχει εξαχθεί από τον μικροεπεξεργαστή PowerPC MPC603e χωρίς τη μονάδα κινητής υποδιαστολής και με power management αλλαγές. Ο πυρήνας είναι υψηλής απόδοσης και χαμηλής ισχύος της οικογένειας PowerPC με το μειωμένο σύνολο εντολών των μικροεπεξεργαστών RISC. Ο πυρήνας του MPC603e εκτελεί το τμήμα των 32 bit της αρχιτεκτονικής του PowerPC, το οποίο παρέχει αποτελεσματικές διευθύνσεις των 32 bit, και ακέραιους τύπους δεδομένων των 8, 16, ανδ 32 bit. Η μνήμη cache του MPC603e παρέχει συνοπινγ για να εξασφαλίσει τη συμβατότητα δεδομένων με άλλους master. Αυτή βοηθά την εξασφάλιση συμβατότητας μεταξύ του CPM και του πυρήνα συστήματος. Ο πυρήνας

Σχήμα 5.1: MPC8260 Block Diagram



περιλαμβάνει 16 Kbytes από instruction cache και 16 Kbytes από data cache. It ηασ ο 64-bit split-transaction εξωτερικό δίαυλο δεδομένων, το οποίο είναι ενωμένο απευθείας στα εξωτερικά pins του MPC8260.

Ο πυρήνας του MPC603e έχει έναν εσωτερικό common on-chip (COP) επεξεργαστή αποσφαλμάτωσης. Αυτός ο επεξεργαστής επιτρέπει την πρόσβαση σε εσωτερική σαρωτικές αλυσίδες για αποσφαλμάτωση. Χρησιμοποιείται επίσης σαν σειριακή σύνδεση με τον πυρήνα για αντίπαλη στήριξη.

Η επίδοση του πυρήνα του MPC603e για το SPEC 95 benchmark για ακέραιες πράξεις εκτείνεται μεταξύ 200 4.4 και 5.1 στα 200 MHz. Στο Dhrystone 2.1 MIPS, ο MPC603e είναι 280 MIPS στα 200 MHz (συγκρινόμενο με το 86 MIPS του MPC860 στα 66 MHz). Ο πυρήνας του MPC603e μπορεί να απενεργοποιηθεί. Σάυτή την κατάσταση ο MPC603e λειτουργεί σαν περιφερικό slave σε έναν εξωτερικό πυρήνα ή σε μια άλλη συσκευή MPC8260 με τον πυρήνα ενεργοποιημένο.

5.3 System Interface Unit (SIU)

Η SIU αποτελείται από τα παρακάτω :

- Ένα 60ξ συμβατό παράλληλο system bus configurable σε εύρος δεδομένων 64-bit. Ο MPC8260 υποστηρίζει μεγέθη πυλών των 64-, 32-, 16-, και 8-bit. Ο εσωτερικός διαιτητής του MPC8620 λειτουργεί μεταξύ εσωτερικών που έχουν πρόσβαση στο βυσ (στον πυρήνα συστήματος, τη CPM, και έναν εξωτερικό master). Ο διαιτητής αυτός μπορεί να απενεργοποιηθεί και, αν είναι απαραίτητο, μπορεί να χρησιμοποιηθεί ένας εξωτερικός διαιτητής.
- Ένα τοπικό δίαυλο (δεδομένων 32 bit, εσωτερικής διεύθυνσης 32 bit, και εξωτερικής 18 bit). Χρησιμοποιείται για την αύξηση της λειτουργίας των ελεγκτών επικοινωνίας πολύ υψηλής ταχύτητας. Χωρίς να απαιτείται εκτεταμένος χειρισμός του πυρήνα, το δίαυλο μπορεί να χρησιμοποιηθεί για να αποθηκεύει πίνακες επικοινωνιών για ATM και περιγραφείς buffer (BD) για να επικοινωνιακά κανάλια ή για τα raw Data που μεταδίδονταν μεταξύ των καναλιών. Το τοπικό bus είναι συγχρονισμένο με το 60ξ bus και τρέχει στην ίδια συχνότητα.
- Ο ελεγκτής μνήμης που υποστηρίζει 12 τράπεζες μνήμης μπορεί να βε αλλοσατεδ είτε για το σύστημα, είτε για το τοπικό bus. Ο ελεγκτής μνήμης είναι μια προηγμένη έκδοση του ελεγκτή μνήμης του MPC860. Υποστηρίζει τρεις μηχανές προγραμματιζόμενες από χρήστες. Εκτός από όλα τα χαρακτηριστικά του MPC860, ο ελεγκτής υποστηρίζει επίσης και SDRAM με page mode και address data pipeline.
- Υποστηρίζει το JTAG controller IEEE 1149.1 test access port (TAP).
- Ένα bus monitor που αποτρέπει κλειδώματα του 60ξ bus, ρολόι πραγματικού χρόνου, χρονιστή περιοδικών διακοπών και άλλες λειτουργίες συστήματος χρήσιμες για ένθετες εφαρμογές.
- Glueless interface με την L2 cache (MPC2605) και 4-/16-K-entry CAM (MCM69C232/MCM69C432).

5.4 Μονάδα Επεξεργαστή Επικοινωνιών (CPM)

Η CPM περιλαμβάνει χαρακτηριστικά που επιτρέπουν στο MPC8260 να υπερτερεί μιας πλειάδας εφαρμογών που στοχεύουν κυρίως σε αγορές δικτύων και τηλεπικοινωνιών. Η CPM είναι ένα υπερσύνολο της CPM MPC860 PowerQUICC, με αυξημένες επιδόσεις και την προσθήκη

hardware και ρουτινών που υποστηρίζουν πρωτόκολλα υψηλού bit-rate, πχ ATM (μέχρι 155 Mbps full-duplex) και Fast Ethernet(100-Mbps full-duplex).

Η παρακάτω λίστα αναφέρει τα κύρια χαρακτηριστικά της CPM:

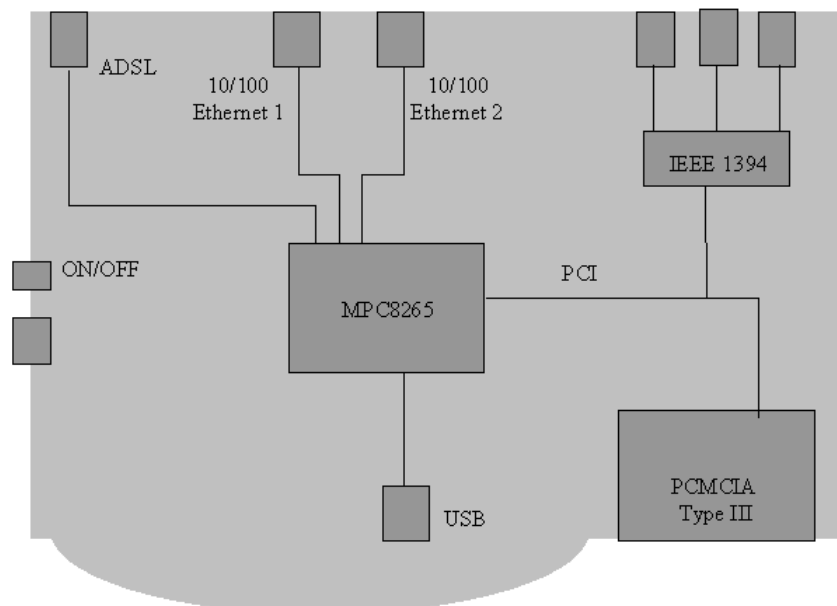
- Ο CP (διαχειριστής επικοινωνιών) είναι ένας ένθετος ελεγκτής RISC 32 bit που βρίσκεται σε διαφορετικό bus (CPM local bus) από το 60x bus (που χρησιμοποιείται από τον πυρήνα του συστήματος). Με αυτό το ξεχωριστό bus, ο CP δεν επηρεάζει την απόδοση του πυρήνα του PowerPC. Ο CP χειρίζεται εργασίες χαμηλών στρωμάτων και δραστηριότητες ελέγχου DMA, επιτρέποντας έτσι στον πυρήνα του PowerPC να ασχοληθεί με υψηλότερου στρώματος δραστηριότητες. Ο CP έχει ένα σύνολο εντολών βελτιστοποιημένων για επικοινωνίες αλλά μπορεί να χρησιμοποιηθεί και για γενικές εφαρμογές, ανακουφίζοντας έτσι το σύστημα από μικρές και συχνά επαναλαμβανόμενες διεργασίες.
- Δύο σειριακές DMA που μπορούν να κάνουν ταυτόχρονες μεταφορές, βελτιστοποιημένες για μεταφορές burst στο 60x bus και το local bus.
- Τρεις full-duplex γρήγορους σειριακούς ελεγκτές επικοινωνιών (FCCs) που υποστηρίζουν πρωτόκολλα ATM (155Mbps) μέσω UTOPIA2 interface (ο MPC8260 έχει δύο UTOPIA ιντερφασες), καθώς και πρωτόκολλα IEEE 802.3 και Fast Ethernet, HDLC μέχρι E3 rates (45 Mbps) και ολοκληρωτικά transparent operation. Κάθε FCC Μπορεί να διαμορφωθεί να μεταφέρει fully transparent και να δεχτεί HDLC ή αντιστρόφως.
- Δύο ελεγκτές πολλαπλών καναλιών (MCCs) που μπορούν να χειριστούν ένα σύνολο από 256×64 Kbps HDLC ή transparent κανάλια, πολυπλεγμένα σε έως και οκτώ TDM interfaces. Οι MCC υποστηρίζουν επίσης υπερ-κανάλια συχνοτήτων υψηλότερων από 64 Kbps Subchanneling απο τα 64-Kbps κανάλια.
- Τέσσερις full-duplex σειριακούς ελεγκτές επικοινωνιών (SCCs) που υποστηρίζουν IEEE802.3/Ethernet, high- level synchronous data link control, HDLC, local talk, UART,synchronous UART, BISYNC, και transparent.
- Δύο full-duplex σειριακούς διαχειριστές επικοινωνιών (SMC) που υποστηρίζουν GCI, UART, και transparent operations.
- Serial peripheral interface (SPI) και I^2C bus controllers
- Time-slot assigner (TSA) που υποστηρίζει πολυπλεξία δεδομένων από κάθε έναν από τους τέσσερις SCCs, τρεις FCCs και δύο SMCs.

Monitor Program

6.1 Ανασκόπηση του InfoStar C1

Το πρώτο μικρό Residential Gateway ήταν το πρωτότυπο InfoStar C1. Το πρωτότυπο αυτό δημιουργήθηκε για την Deuche Telecom. Με κύριο Επεξεργαστή Μοτορολα MPC8265, το InfoStar C1 φτιάχτηκε για να προσφέρει Υπηρεσίες Ethernet, Firewire, PCMCIA, USB, καθώς και τηλεφωνική υποστήριξη. Το διάγραμμα αυτού του πρωτοτύπου φαίνεται στην εικόνα 6.1.

Σχήμα 6.1: InfoStar C1 Block Diagram



Το InfoStar C1, ως μέλος της οικογένειας InfoStar Compact, έχει σχεδιαστεί με το σκεπτικό της ελάχιστης διαχείρισης και της ικανότητας για plug-n-play. Τρέχει Linux OS, οπότε παρέχει ανοιχτή αρχιτεκτονική λογισμικού και μια πλατιά ποικιλία από drivers, πρωτόκολλα και υπηρεσίες.

Ένας μικρός κατάλογος των υπηρεσιών περιλαμβάνει Web server, DNS, DHCP, NAT και Firewall, ενώ το σύστημα είναι ανοιχτό σε κάθε όριμο ή innovative πρωτόκολλο, υπηρεσία, ή εφαρμογή. Με την ελαστικότητα να παρουσιάζει νέα interface μέσω της PCMCIA interface, το InfoStar/C1 στοχεύει σε νέα απαιτητικά περιβάλλοντα στο σπίτι και στα περιβάλλοντα SOHO με μέσο σύνδεσης το τηλεφωνικό καλώδιο.

Η μνήμη Flash χρησιμοποιήθηκε για να αποθηκεύει τον πυρήνα και το σύστημα αρχείων του πίνακα. Επίσης το MPC8265 προσφέρει υποστήριξη PCI, ώστε να μπορούμε να ελέγχουμε και το PCMCIA και το Firewire.

Όταν πήραμε την πλακέτα από το εργαστήριο, έπρεπε να κάνουμε import ένα monitor πρόγραμμα, καθώς και ένα μικρό σύστημα αρχείων με τον πυρήνα του Linux.

6.2 Ανασκόπηση του Monitor προγράμματος

Αρχικοποίηση του συστήματος

Κατα την διαδικασία του reset, το PPCBoot ξεκινάει στη διεύθυνση 0x00000100. Λόγο της διαμόρφωσης του reset για τοCS0, αυτό είναι Ένας καθρέφτης στην Flash μνήμης του συστήματος. Για να μπορούμε να είμαστε ικανοί να κάνουμε re-map την μνήμη, το PPCboot κάνει jump στην link address. Για να μπορέσουμε να γράψουμε τον αρχικό κώδικα σε C, πρέπει να αρχικοποιήσουμε μια μικρή στοίβα στην εσωτερική Dual Ported RAM του MPC8xxx. Ύστερα από αυτό, το PPCBoot αρχικοποιεί το CPU core, τις caches και το SIU.

Στη συνέχεια, όλες οι ελεύθερες memory banks γίνονται mapped χρησιμοποιώντας ένα προκαταρκτικό mapping, το οποίο τις βάζει σε όρια των 512 MB, (πολλαπλάσια του 0x20000000: SDRAM on 0x00000000 and 0x20000000, Flash on 0x40000000 and 0x60000000, SRAM on 0x80000000). Στη συνέχεια ο UPM προγραμματίζεται για SDRAM access. Χρησιμοποιώντας την προχειρή αυτή σύνθεση, γίνεται ένα απλό τεστ, για να βρούμε το μέγεθος της εξωτερικής δυναμικής μνήμης.

Όταν υπάρχουν περισσότερες από μία τράπεζες SDRAM και είναι όλες διαφορετικού μεγέθους σχεδιάζεται πρώτα η μεγαλύτερη. Για τράπεζες ίδιου μεγέθους σχεδιάζεται πρώτα η πρώτη (CS2). Η πρώτη σχεδίαση είναι πάντα για τη διεύθυνση 0x00000000, με οποιοδήποτε συμπληρωματικές τράπεζες να ακολουθούν ώστε να δημιουργήσουν μια γειτονική μνήμη ξεκινώντας από το 0.

Ύστερα, η οθόνη αυτοεγκαθίσταται στο πάνω άκρο της SDRAM και δεσμεύει μνήμη για

χρήση και για global Board info data, με τη malloc(). Επίσης ο exception vector code αντιγράφεται στις χαμηλές σελίδες της RAM και στήνεται η τελική στοιβα.

Μόνο μετά το relocation έχουμε ένα φυσιολογικό περιβάλλον C. Μέχρι τότε υπάρχουν πολλών ειδών περιορισμοί, κυρίως διότι τρέχουμε μέσω της ROM και επειδή ο κώδικας πρέπει να γίνει relocate σε μια νέα διεύθυνση της RAM.

Διαχείριση μνήμης

Το PPCBoot τρέχει σε κατάσταση συστήματος και χρησιμοποιεί φυσικές διευθύνσεις. Για παράδειγμα το MMU δεν χρησιμοποιείται ούτε για καθορισμό διευθύνσεων ούτε για προστασία μνήμης.

Η διαθέσιμη μνήμη σχεδιάζεται για να φτιάχνει διευθύνσεις με χρήση του ελεγκτή μνήμης. Σ' αυτή τη διεργασία σχηματίζεται ένα block για κάθε τύπο μνήμης (Flash SDRAM, SRAM), ακόμα κι αν συνίσταται από πολλές φυσικές τράπεζες μνήμης.

Το PPCBoot είναι εγκατεστημένο στα πρώτα 128 kB της πρώτης τράπεζας Flash(στο TQM8xxL αυτό είναι το πεδίο 0x40000000 ... 0x4001FFFF). Μετά την εκκίνηση και την αρχικοποίηση της DRAM, ο κώδικας βρίσκεται αυτόματα στο πάνω άκρο της DRAM. Ακριβώς κάτω από τον κώδικα PPCBoot φυλάγεται κάποιος χώρος προς χρήση με την malloc(). Κάτω από αυτόν, βρίσκεται μια δομή με global Board Info data, ακολουθούμενη από μια στοιβα (που μεγαλώνει προς τα κάτω).

Επιπλέον, ένας exception handler code αντιγράφεται στα κάτω 8 kB της DRAM (0x00000000 ... 0x00001FFF). Οπότε ένα τυπικό ζονφιγουρατιον με 16 MB DRAM είναι ως εξής:

```
0x0000 0000      Exception Vector code
      :
0x0000 1FFF
0x0000 2000      Free for Application Use
      :
      :
      :
      :
0x00FB FF20      Monitor Stack (Growing downward)
0x00FB FFAC      Board Info Data
0x00FC 0000      Malloc Arena
      :
0x00FD FFFF
```

```
0x00FE 0000    RAM Copy of Monitor Code
0x00FF FFFF    [End of RAM]
```

Περιβάλλον

Το PPCBoot υποστηρίζει user configuration χρησιμοποιώντας Environment Variables που μπορούν να διατηρηθούν με το να γραφτούν στη μνήμη Flash.

Οι μεταβλητές περιβάλλοντος ρυθμίζονται με "setenv", τυπώνονται με "printenv" και σώζονται στη Flash με "saveenv". Η setenv χωρίς τιμή χρησιμοποιείται για να διαγραφεί μια μεταβλητή από το περιβάλλον. Όσο δεν σώζεται το περιβάλλον η δουλειά γίνεται με in-memory copy. Σε περίπτωση που ο χώρος της Flash που περιέχει το περιβάλλον σθησει κατά λάθος, παρέχεται ένα περιβάλλον default.

Παρόλο που το PPCBoot θα πρέπει να υποστηρίζει εύκολα οποιαδήποτε εφαρμογή OS ή standalone, η βασική μέριμνα κατά το σχεδιασμό του PPCBoot ήταν το Linux.

Το PPCBoot περιέχει πολλά στοιχεία, που μέχρι τώρα ήταν μέρη ενός ειδικού κώδικα "boot loader" μέσα στον πυρήνα του Linux. Επίσης οποιαδήποτε εικόνα "initrd" χρησιμοποιείται, δεν είναι πια κομμάτι μια μεγάλης εικόνας του Linux. Αντίθετα πυρήνας και initrd είναι διαφορετικές εικόνες. Αυτός ο διαχωρισμός εξυπηρετεί τους εξής σκοπούς:

- Τα ίδια features μπορούν να χρησιμοποιηθούν για άλλες εφαρμογές OS ή standalone (για παράδειγμα η χρήση συμπιεσμένων εικόνων για να μειωθεί το footprint της μνήμης Flash).
- Είναι πολύ πιο εύκολο να κάνεις port καινούργιες εκδόσεις του πυρήνα του Linux γιατί πολλές χαμηλού επιπέδου εργασίες, εξαρτώμενες από το hardware, γίνονται από το PPCBoot.
- η ίδια εικόνα του πυρήνα του Linux μπορεί να χρησιμοποιηθεί με διαφορετικές εικόνες initrd. Βέβαια, αυτό σημαίνει επίσης ότι διαφορετικές εικόνες του πυρήνα μπορούν να τρέξουν με την ίδια initrd. Έτσι ο έλεγχος γίνεται ευκολότερος (δεν χρειάζεται να δημιουργηθεί καινούργια εικόνα "zImage.initrd", κάθε φορά που αλλάζει ένα αρχείο στο "initrd"). Επίσης ευκολότερο γίνεται και το πεδίο αναβάθμισης του λογισμικού.

6.3 Διαδικασία Import

Το μόνο εργαλείο που είχαμε στη διάθεσή μας για να δοκιμάσουμε και να προγραμματίσουμε την προτύπη πλακέτα, ήταν το JTAG. Είχαμε στην διάθεσή μας τον OCDCommander της Macgraicor, καθώς επίσης και την διασύνδεση Parallel Reaver Blackbird to JTAG.

Υστερα απο την επιβεβαίωση οτι ο επεξεργαστής λειτουργούσε σωστά, (διαμέσου του σήματος CS0_, το οποίο ανέβαινε και κατέβαινε συνεχώς), έπρεπε να γράψουμε στη μνήμη flash, την Hard Reset Configuration Word. Για να το επιτύχουμε, έπρεπε να διαβάσουμε το εγχειρήδιο της μνήμης και να καταλάβουμε τον τρόπο με τον οποίο έπρεπε να προγραμματιστεί η μνήμη αυτή. Το καλό με το εργαλείο JTAG, ήταν οτι μπορούσαμε να γράψουμε σε οποιαδήποτε φυσική μνήμη του συστήματος θέλαμε. Έτσι, μπορούσαμε να γράψουμε στους εσωτερικούς καταχωρητές του επεξεργαστή 8265, όπως επίσης και την ίδια την flash. Η HRCW επιλέχτηκε με προσοχή, μιας και είναι υπεύθυνη για την σωστή αρχικοποίηση και σωστή λειτουργία του επεξεργαστή. Ακόμα στην HRCW δηλώνουμε και το πλάτος του διαυλου της δυναμικής μνήμης, όπως επίσης και το πλάτος της flash μνήμης.

Αφού τελειώσαμε με το βήμα αυτό, έπρεπε να είμαστε σίγουροι πως ο επεξεργαστής μπορούσε να εκτελεί σωστά τις εντολές. Έτσι αναγκαστήκαμε να γράψουμε μικρά προγράμματα σε γλώσσα μηχανής που να γράφουμε μερικούς εσωτερικούς καταχωρητές ή ακόμα και την εξωτερική δυναμική μνήμη. Μπορούσαμε ακόμα να γράψουμε και σε κάποιους καταχωρητές που ήταν υπεύθυνοι για τις δύο κάρτες δικτύου. Έτσι μπορούσαμε να ανάβουμε και να σβήνουμε τα λαμπάκια των καρτών δικτύου μέσα από τα προγράμματά μας.

Όσο όμως τα προγράμματα που θέλαμε να γράψουμε στην flash μεγαλώνανε, χρειαζόμασταν ένα πρόγραμμα που θα μετέτρεπε το δυαδικό αρχείο του μεταγλωτιστή σε ένα αρχείο MAC του OCD Commander, το οποίο θα καταλάβαινε και θα εκτελούσε.

Το αρχείο που δημιουργούσε ο μεταγλωτιστής, ήταν σε SREC format, το οποίο το είχε ορίσει η ίδια η Motorola. Έτσι έπρεπε να διαβάσουμε αυτό το αρχείο του μεταγλωτιστή, και να δημιουργήσουμε ένα απλό αρχείο κειμένου το οποίο θα όριζε τι δεδομένα θα έπρεπε να γραφτούν σε τί διευθύνσεις μνήμης. Για παράδειγμα, χρειαζόμασταν ένα αρχείο που θα περιείχε τον ακόλουθο τύπο γραμμών.

```
word 0xFE002204 = 0xFFEDE122
```

Έτσι, αναγκαστήκαμε να γράψουμε ένα πρόγραμμα σε C, το οποίο θα έκανε αυτή τη μετατροπή. Αυτό το πρόγραμμα, παρά το γεγονός οτι φαινόταν πολύ ευκολο, αποδείχτηκε πως ήταν ένα πρόγραμμα 500 γραμμών. Για την μεταγλώτιση του προγράμματος, χρησιμοποιήσαμε την Visual C++ της Microsoft.

6.3.1 PPCBoot 1.0.1 loader

Το επόμενο βήμα ήταν να κάνουμε import το PPCBOOT monitor program. Έτσι κατεβάσαμε απο το διαδίκτυο το ppboot-1.0.1 bootloader και ξεκινήσαμε να το μελετάμε, για να καταλάβουμε τον τρόπο με τον οποίο λειτουργεί.

Έπρεπε να τροποποιήσουμε το , έτσι ώστε να υποστηρίζει το δικό μας προτότυπο. Η όλη διαδικασία έγινε ακόμα πιο δύσκολη, από το γεγονός πως χρειαζόμασταν περίπου μισή ώρα για να προγραμματίσουμε το rreboot στην flash. Έτσι οι δοκιμές μας για μία μέρα ήταν περιορισμένες σε έξι με επτά προσπάθειες.

Έτσι μάθαμε όσα χρειαζόμασταν για την αρχικοποίηση του επεξεργαστή, και σε πια σημεία χρειαζόταν να κάνουμε αλλαγές. Αλλά το monitor προγραμμα, δεν φαινόταν να τρέχει σωστά. Μπορούμε εύκολα να καταλάβουμε πως μιας και δεν είχαμε κάποιο προφανή τρόπο για debugging (πχ, printk), η δουλειά μας ήταν πολύ δύσκολη.

Το monitor program ήταν γραμμένο στην μνήμη flash, και όταν ξεκίναγε να τρέχει αντέγραφε τον εαυτό του στην δυναμική μνήμη, και μετά εκτελούσε όλες τις εντολές από την δυναμική μνήμη. Μιας και το InfoStar C1 ήταν ένα προτότυπο, τα πάντα ήταν δυνατό. Το πρόβλημα μπορούσε να ήταν τόσο στο λογισμικό όσο και στο hardware.

Υστερα από πολλές μέρες debugging, αλλάξαμε το σημείο στο οποίο αντεγραφόταν το πρόγραμμα στη μνήμη, και μπορέσαμε και πήραμε το πρώτο αποτέλεσμα στη σειριακή. Αυτό έκανε το debugging πολύ πιο εύκολο.

6.3.2 Flash Driver

Στη συνέχεια, έπρεπε να υπερνικήσουμε πολλά προβλήματα. Το πρώτο ήταν ο οδηγός για την μνήμη flash. Το rreboot ήταν σχεδιασμένο να δουλεύει με 32bit flash μνήμες. Στο δικό μας πρωτότυπο, η μνήμη flash είχε διάυλο πλάτους 64 bits. Έτσι αναγκαστήκαμε να αλλάξουμε τον οδηγό για την μνήμη αυτή. Εδώ, χρειάστηκε να αλλάξουμε περίπου 500 γραμμές κώδικα μέσα στο boot loader.

Το αρχείο που αλλάχτηκε, ήταν το flash.c μέσα στη δομή των καταλόγων του rreboot. Η flash_init ήταν η πρώτη συνάρτηση που έπρεπε να αλλάξει. Μιας και το πλάτος του διαύλου μας ήταν διπλάσιο, έπρεπε να στέλνουμε 64bit εντολή, κάθε φορά που θέλαμε να προσπελάσουμε την μνήμη. Έτσι έπρεπε πολλές μεταβλητές και εντολές να αλλάζουν από word σε dword. Ομοίως και οι εντολές.

Στη συνέχεια, οι δύο πιο κύριες συναρτήσεις που έπρεπε να αλλάξουν, ήταν οι write_buff, write_word. Η δυσκολία με τις αλλαγές αυτές, ήταν το word allignment, και ο τρόπος που έπρεπε να χρησιμοποιούμε τα unaligned tail bytes.

Αφού τελειώσαμε το βήμα αυτό, πήραμε prompt, αλλά δεν μπορούσαμε να λάβουμε καμιά απάντηση από την πλακέτα. Έμοιαζε, λες και δεν λάμβανε καθόλου δεδομένα. Έτσι φτάσαμε στο συμπέρασμα πως κάτι γίνεται με την σειριακή. Το πρόβλημα, τελικά λύθηκε, και είχαμε πια το rreboot boot loader να τρέχει στο σύστημά μας.

6.4 Ethernet Support Configuration

Το επόμενο βήμα ήταν να φτιάξουμε την υποστήριξη για την κάρτα δικτύου του ppcboot. Αυτό ήταν πραγματικά πολύ απαραίτητο, μιας και μέσω της ethernet μπορούσαμε να κατεβάσουμε πυρήνα του linux πολύ γρήγορα. Είναι φανερό πως δεν μπορούσαμε να χρησιμοποιήσουμε το jtag για να κατεβάσουμε έναν πυρήνα. Δεν είχαμε την άνεση του χρόνου για κάτι τέτοιο.

Το ρύθμιση για την ethernet, δεν ήταν τόσο δύσκολο. Το μόνο πράγμα που χρειάστηκε να κάνουμε, ήταν να κατανοήσουμε σε βάθος την διαφορά ανάμεσα στο δικό μας προτότυπο και στο sbc8260. Οι βασικές διαφορές ήταν τα pin assignments και το memory map. Ετσι έπρεπε να αλλάζουμε δύο configuration αρχεία, και ήμασταν ποια έτοιμοι για να δοκιμάσουμε την κάρτα δικτύου. Η μικρή αυτή τροποποίηση πήρε περίπου τρεις μέρες, και αλλάξαμε περίπου 50 γραμμές κώδικα.

Linux Kernel

7.1 Linux Kernel Version

Μιας και θέλαμε να υποστηρίξουμε ταυτόχρονα και το FireWire, αλλά και το ATM, η μόνη μας επιλογή ήταν ο πυρήνας linux 2.4.x. Αυτή η έκδοση του πυρήνα, όχι μόνο υποστήριζε το H323 FireWire αλλά και υπηρεσίες Quality of Service για το ATM. Όταν ξεκινήσαμε την διαδικασία της ενσωμάτωσης του πυρήνα, η έκδοση που υπήρχε ήταν η 2.4.7.

Η μόνη βοήθεια που είχαμε, ήταν η πλακέτα ανάπτυξης της Motorola, MPC8260 VADS. Επίσης είχαμε και την πλατφόρμα ανάπτυξης λογισμικού της BlueCat. Η εταιρία αυτή παρείχε μια ολοκληρωμένη πλατφόρμα linux για την πλακέτα ανάπτυξης της Motorola. Το πρόβλημα όμως ήταν στο ότι ο πυρήνας του linux που υποστήριζε ήταν προηγούμενης γενιάς. Ο πυρήνας αυτός ήταν τόσο παλιός, ώστε να μην παρέχει υπηρεσίες Quality of Service. Έτσι έπρεπε να μεταβούμε σε νέα έκδοση του λειτουργικού. Οι δύο αυτοί πυρήνες είχαν τρομερές διαφορές μεταξύ τους. Ειδικά στον τομέα της διαχείρισης της μνήμης. Έτσι δεν είχαμε και μεγάλη βοήθεια από τον παλιό πυρήνα.

7.2 Debugging method

Όταν φορτώσαμε τον πρώτο πυρήνα, δεν είχαμε καμμία έξοδο στην κονσόλα. Το debugging ήταν τρομερά δύσκολο. Το μόνο που μπορούσαμε να κάνουμε, ήταν διαμέσου του JTAG να δούμε σε ποια διεύθυνση σταμάταγαν να εκτελούντε οι εντολές του πυρήνα. Στη συνέχεια, συμβουλευόμασταν το αρχείο System.map, για να βρούμε σε ποια συνάρτηση σταμάταγε η εκτέλεση των εντολών. Βέβαια, μιας και στα πρώτα στάδια του, ο κώδικας του πυρήνα ήταν σε assembly, δεν μπορούσαμε να βρούμε τι ήταν αυτό που πραγματικά έφταιγε.

Το πρώτο πράγμα που έπρεπε να ελέγξουμε, ήταν η δομή που board_info_t. Όταν το monitor πρόγραμμα ξεκινάει την εκτέλεση του πυρήνα, του περνάει μια δομή με πληρο-

φορίες για το σύστημα. Η δομή αυτή, έχει πληροφορίες για το ρολόι του επεξεργαστή, του δίαυλου, του filesystem, της κάρτας δικτύου και αρκετά ακόμη. Άρα έπρεπε να ελέγξουμε αν η πληροφορία που περνάγαμε στον πυρήνα ήταν σωστή.

Ομως η πληροφορία αυτή δεν ήταν σωστή. Έπρεπε να μεταβάλουμε τη δομή. Την αλλαγή την κάναμε στο `ppcboot`, μιας και θέλαμε να αλλάξουμε τον πυρήνα όσο το δυνατό λιγότερο. Όταν ολοκληρώσαμε με την αλλαγή της δομής αυτής, μπορέσαμε και πήραμε ένα είδος μηνύματος στη σειριακή, αλλά δεν μπορούσαμε να τα διαβάσουμε. Το επόμενο βήμα ήταν ο οδηγητής της σειριακής κονσόλας.

Ο πυρήνας ήταν φτιαγμένος για να αρχικοποιεί την κονσόλα στο SMC (Serial Management Controller) και όχι στο SCC, που είχαμε εμείς ενσωματωμένη την σειριακή μας. Έτσι έπρεπε να αλλάξουμε τον οδηγητή του πυρήνα για τη συγκεκριμένη συσκευή. Ευτυχώς, δεν χρειάστηκε να αλλάξουμε τον κώδικα, μιας και βρήκαμε ένα μικρό patch στο διαδίκτυο που διόρθωνε το πρόβλημα αυτό. Ύστερα από την εκτέλεση του patch, μπορούσαμε να διαβάζουμε τα μηνύματα από την σειριακή.

Αλλά ο πυρήνας πάλι δεν δουλευε σωστά. Σταμάταγε όταν προσπαθούσε να κάνει `map` μία εξωτερική συσκευή στη διεύθυνση `0x80000000`. Έτσι αναγκαστήκαμε να αφαιρέσουμε την αρχικοποίηση αυτή (μιας και δεν την χρειαζόμασταν), και ο πυρήνας ξεκίνησε με επιτυχία. Το επόμενο βήμα, ήταν το filesystem. Την πρώτη φορά αντι για filesystem χρησιμοποιήσαμε ένα `hello_world` πρόγραμμα. Αυτό έτρεχε σωστά. Όταν όμως δοκιμάσαμε ένα σχετικά μικρό filesystem, ο πυρήνας σταμάταγε να δουλεύει.

Πάλι ξανά δεν είχαμε κάποιο μήνυμα στην κονσόλα. Όταν ο πυρήνας προσπαθούσε να κάνει `mount` το σύστημα αρχείων, απλά σταμάταγε να εκτελεί εντολές, χωρίς να μας αφήνει κάποιου είδους μηνύματα. Ύστερα από αρκετές μέρες `debugging`, δημιουργήσαμε ένα μικρό πρόγραμμα σε γλώσσα μηχανής που έλεγχε τη μνήμη. Έτσι μπορέσαμε και βρήκαμε το πρόβλημα. Η μνήμη είχε κάποιο πρόβλημα, και μερικά δεδομένα γραφόντουσαν σε περισσότερες απο μία διευθύνσεις στη μνήμη. Έτσι ξεκινήσαμε να ψάχνουμε το πρόβλημα στο hardware.

Έτσι, μπορέσαμε στο τέλος να βρούμε την αιτία του προβλήματος μας. Ένα ποδαράκι από το ένα από τα τέσσερα ολοκληρωμένα της μνήμης, δεν έκανε καλή επαφή με το τυπωμένο κύκλωμα. Έτσι μία `address line`, δεν ήταν ενωμένη με ένα απο τα ολοκληρωμένα.

Ύστερα από αυτό, όλα πήγαν καλά. Μπορούσαμε πια να κατεβάσουμε ένα μικρό file system, και να αναγνωρίσουμε τις δύο κάρτες δικτύου. Έτσι μπορέσαμε να χρησιμοποιήσουμε και το NFS filesystem.

ATM Interface

8.1 Επικοινωνιακή Μονάδα Επεξεργαστή

Η επικοινωνιακή μονάδα επεξεργαστή MPC8260 (CPM) είναι ένα υπερσύνολο της CPM MPC860 PowerQUICC, με αυξημένες επιδόσεις και την προσθήκη hardware και ρουτινών για την υποστήριξη πρωτοκόλων υψηλού bit-rate (πχ ATM και Fast Ethernet). Η υποστήριξη πολλαπλών καναλιών HDLC έχει προαχθεί έτσι ώστε να υποστηρίζει μέχρι 256 κανάλια. Στην εικόνα 8.1, φαίνεται το διάγραμμα της Επικοινωνιακής Μονάδα Επεξεργαστή.

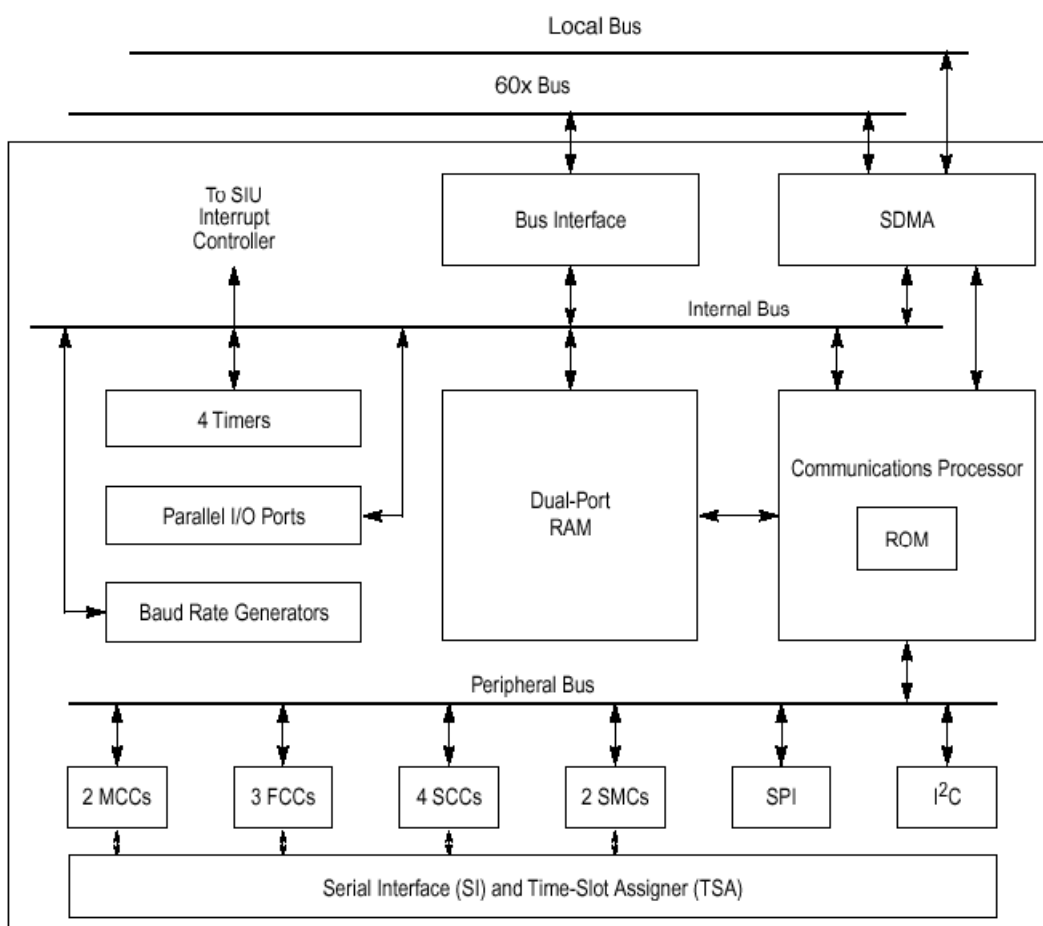
8.1.1 Χαρακτηριστικά

Η CPM περιλαμβάνει διάφορα block ώστε να μπορεί το σύστημα να χειρίζεται αποδοτικά εργασίες επικοινωνίας δεδομένων. Τα παρακάτω είναι μια λίστα των σημαντικών χαρακτηριστικών της CPM.

- Επεξεργαστής Επικοινωνιών (CP)
 - Μία εντολή ανα κύκλο ρολογιού
 - Εκτελεί κώδικα από εσωτερική ROM ή dual port RAM
 - Αρχιτεκτονική RISC 32 bit
 - Εσωτερικός χρονιστής
 - Interfaces με τον ένθετο πυρήνα του επεξεργαστή μέσω μιας δυαδικής πύλης RAM 24 bit και εικονικών καναλιών DMA για κάθε περιφερικό ελεγκτή.
- Τέσσερις full-duplex σειριακούς ελεγκτές επικοινωνιών (SSCs)
- Δύο full-duplex σειριακούς διαχειριστές επικοινωνιών (SMCs)

- Τρεις full-duplex γρήγορους σειριακούς ελεγκτές επικοινωνιών (FCCs)
- Δύο ελεγκτές πολλαπλών καναλιών (MCCs) που μπορούν μαζί να χειριστούν μέχρι και 256 HDCL ανά transparent channels 64 Kbps το καθένα, πολυπλεγμένα μέχρι και σε οκτώ TDM interfaces
- Time-slot assigner υποστηρίζει πολυπλεξία των δεδομένων απο οποιαδήποτε SCCs, FCCs, SMCs, και MCCs σε οκτώ time-division πολυπλεγμένα (TDM) interfaces.
- Οκτώ ανεξάρτητους baud rate generators (BRGs)
- General-purpose parallel ports και 16 παράλληλες γραμμές εισόδου/εξόδου με ικανότητα διακοπής

Σχήμα 8.1: MPC8260 CPM Block Diagram



8.2 Διαχειριστής Επικοινωνιών (CP)

Ο διαχειριστής επικοινωνιών (CP), ή αλλιώς ο μικροδιαχειριστής RISC, είναι ένας διαχειριστής 32 bit για τη CPM που βρίσκεται σε ξεχωριστό bus από τον πυρήνα και επομένως μπορεί να εργάζεται ανεξάρτητα από τον πυρήνα του PowerPC. Ο CP διαχειρίζεται επικοινωνιακές εργασίες χαμηλών στρωμάτων, επιτρέποντας έτσι στον πυρήνα να ασχοληθεί με υψηλότερου στρώματος δραστηριότητες. Ο CP δουλεύει με τους περιφερειακούς διαχειριστές και τις παράλληλες πύλες για να εκτελέσει user-programmable protocols και να διευθύνει σειριακά κανάλια DMA (SDMA) που μεταφέρουν πληροφορίες μεταξύ των καναλιών εισόδου/εξόδου και της μνήμης. Επίσης διευθύνει και τα κανάλια IDMA (ανεξάρτητη DMA) και περιέχει εσωτερικό χρονιστή που χρησιμοποιείται για να implement up to 16 επιπρόσθετους χρονιστές.

Η αρχιτεκτονική και το σύνολο οδηγιών του CP είναι optimized για επικοινωνίες και επεξεργασίες δεδομένων που απαιτούνται από πολλά καλωδιακά και μη επικοινωνιακά πρότυπα.

8.2.1 Χαρακτηριστικά

Τα παρακάτω είναι μια λίστα των σημαντικών χαρακτηριστικών του CPM.

- Ένας κύκλος χρονιστή του συστήματος ανά οδηγία
- 32-bit instruction object code
- Εκτελεί κώδικα από εσωτερική ROM ή RAM
- 32-bit ALU data path
- 64-bit dual-port RAM access
- Optimized for communications processing
- Performs DMA bursting of serial data from/to dual-port RAM to/from external memory

8.2.2 CP Block Diagram

Ο CP περιέχει τις ακόλουθες λειτουργικές μονάδες:

- Scheduler και sequencer
- Αποκωδικοποιητή οδηγιών
- Εκτελεστική μονάδα

- Μονάδα φόρτωσης/αποθήκευσης (LSU)
- Μονάδα μεταφοράς block (BTM) - moves data between serial FIFO and RAM
- Οκτώ καταχωρητές γενικού σκοπού (GPRs) Ειδικό καταχωρητές, μηχανή CRC, HDLC framer

8.3 Ανασκόπηση του ελεγκτή του ATM

Ο ελεγκτής ATM παρέχει τα στρώματα ATM και AAL του πρωτοκόλου ATM χρησιμοποιώντας τον διεθνή έλεγχο και operations physical layer (PHY) interface για το ATM (UTOPIA level II) και για τα δύο master και slave modes. Εκτελεί λειτουργίες segmentation and reassembly (SAR) των AAL5, AAL1, και AAL0, και των περισσότερων κοινών μερών του convergence υποστρώματος (CP-CS) αυτών των πρωτοκόλων.

Για κάθε εικονικό κανάλι (VC), ο ελεγκτής της μονάδας βηματικού ελέγχου του ATM (APC) παράγει ένα cell transmission rate για να εκτελέσει συνεχές bit rate (CBR), διαφοροποιούμενο bit rate (VBR), διαθεσιμο bit rate (ABR), ακαθόριστο bit rate (UBR) ή UBR+ traffic. Για να ρυθμιστεί το VBR traffic, η μονάδα του APC εκτελεί έναν συνεχή αλγόριθμο του στάζοντος κουβά. Χρησιμοποιεί επίσης μέχρι 8 επίπεδα προτεραιότητας για να θέσει τα real-time ATM channels, όπως τα CBR και time VBR, πιο πάνω σε ιεραρχία από τα non-real-time ATM channels όπως τα VBR, ABR και UBR.

Ο ελεγκτής ATM διενεργεί το ATM Forum (UNI-4.0) ABR flow control.

Ο MPC8260 υποστηρίζει μια ειδική κατάσταση για ATM/TDM interworking. Η CPM εκτελεί αυτόματα την προώθηση δεδομένων μεταξύ καναλιών ATM και MCCs, TDM χωρίς την παρεμβολή του πυρήνα.

8.3.1 Ανασκόπηση του Transmitter

Πριν τεθεί σε λειτουργία ο μεταδότης, ο host πρέπει να αρχικοποιήσει το MPC8260 και να δημιουργήσει την transmit data structure ATM Memory Structure. Όταν τα δεδομένα είναι έτοιμα για μετάδοση ο host ταχτοποιεί τον πίνακα BD και γράφει το δείκτη του πρώτου BD στον transmit connection table (TCT). Ο host εκδίδει μια εντολή **ATM TRANSMIT**, η οποία εισάγει το τρέχον κανάλι στη μονάδα βηματικού ελέγχου του ATM. Η μονάδα APC διευθύνει την κυκλοφορία του ATM του μεταδότη. Διαβάζει τις παραμέτρους της κυκλοφορίας του κάθε καναλιού και διαιρεί το ολικό bandwidth μεταξύ τους. Η μονάδα APC μπορεί να εντοπίζει το peak cell rate, peak-and-sustain cell rate (GCRA traffic) ή το peak-and-minimum cell rate traffic. Το APC εκτελεί μέχρι 8 επίπεδα προτεραιότητας για την υπηρεσία πρώτα των καναλιών real-time και μετά των non-real-time.

Ο transmitter ATM cell είναι 53-65 bytes και περιλαμβάνει 4 bytes ATM cell header, ένα HEC 1 byte, και 48 bytes payload. Το HEC είναι μια σταθερά λαβανομένη από το FDSRx[0-15] όταν χρησιμοποιούμε UTOPIA 16 και από το FDSRx[0-15] όταν χρησιμοποιούμε UTOPIA 8, FCC Data Synchronization Registers (FDSRx). Τα cell που ορίζονται από το χρήστη (UDC) περιλαμβάνουν επιπλέον header 1 - 12 bytes με προαιρετικό HEC octet. Οι μεταφορές πακέτων χρησιμοποιούν UTOPIA level II, cell-level handshake.

Η μετάδοση αρχίζει όταν το APC προγραμματίζει ένα κανάλι. Σύμφωνα με τον κώδικα του καναλιού, ο ελεγκτής ATM διαβάζει την είσοδο του καναλιού στο TCT και ανοίγει το πρώτο BD για μετάδοση.

8.3.2 Ανασκόπηση του AAL5 Transmitter

Ο μεταδότης διαβάζει 48 bytes από τον εξωτερικό ενταμιευτή, προσθέτει το cell header και στέλνει το cell μέσω του UTOPIA interface. Ο μεταδότης προσθέτει ό,τι παδδινγ χρειάζεται και προσθέτει το AAL5 trailer στο τελευταίο cell του AAL5 frame. Το trailer συνίσταται από CPCS-UU+CPI, μήκος δεδομένων και CRC-32 όπως ορίζεται στο ITU I.363. Το CPCS-UU+CPI (με είσοδο 2 byte) μπορεί να καθορισθεί από το χρήστη ή να διαγραφεί προαιρετικά από τον μεταδότη στο Transmit Connection Table (TCT). Ο μεταδότης αναγνωρίζει το τελευταίο cell του μηνύματος του AAL5 θέτοντας το τελευταίο (L) indication bit στο πεδίο PTI του cell header. Μπορεί να γίνει μια διακοπή για να δείξει το τέλος του frame. Όταν λήξει η μετάδοση του τρέχοντος frame και δεν υπάρχουν στον πίνακα πρόσθετοι έγκυροι ενταμιευτές, η διαδικασία μετάδοσης τελειώνει. Ο μεταδότης συνεχίζει τους επαναλαμβανόμενους ελέγχους του πίνακα BD κάθε φορά που αυτό το κανάλι είναι προγραμματισμένο να μεταδώσει. Στην κατάσταση auto-VC-off το APC απενεργοποιεί αυτόματα το τρέχον κανάλι όταν κανένας ενταμιευτής δεν είναι έτοιμος να μεταδώσει. Στην περίπτωση αυτή χρειάζεται μια νέα εντολή **ATM TRANSMIT** για να ξανααρχίσει η μετάδοση του VC. Σημειώστε ότι κατά τη διάρκεια της μετάδοσης frame η υπόδειξη buffer-not-ready σταματάει τη μεταφορά.

8.3.3 Ανασκόπηση του Receiver

Πριν ενεργοποιηθεί ο μεταδότης, ο host πρέπει να αρχικοποιήσει τον MPC8260 και να δημιουργήσει τη δομή δεδομένων αποδοχής, την ATM Memory Structure. Ο host σχεδιάζει έναν πίνακα BD για κάθε κανάλι ATM. Οι ενταμιευτές κάθε σύνδεσης μπορούν να δεσμευτούν στατικά (αν κάθε BD στον πίνακα BD είναι συσχετισμένο με ένα συγκεκριμένο μέρος του ενταμιευτή) ή, στην περίπτωση του AAL5, μπορούν να γίνουν fetched από τον CP από το global free buffer pool. Το μέγεθος του cell του δέκτη ATM είναι 53-65 byte. Το cell περιλαμβάνει: 4 bytes ATM cell header, 1 byte HEC, που παραβλέπεται, και 48 bytes payload. Τα cell που ορίζονται από το χρήστη (UDC) περιλαμβάνουν επιπλέον header 1 -

12 bytes με προαιρετικό HEC octet. Οι μεταφορές των πακέτων χρησιμοποιούν το UTOPIA level II, cell-level handshake.

Η αποδοχή αρχίζει όταν το PHY θέτει το διαθέσιμο σήμα του cell αποδοχής να υποδεικνύει ότι το PHY έχει ένα ολοκληρωμένο cell στη FIFO υποδοχής του. Ο δέκτης διαβάζει ένα ολοκληρωμένο cell από το UTOPIA interface και μεταφράζει την τη header address (VP/VC) σε έναν κώδικα καναλιού εκτελώντας μια αναζήτηση στις διευθύνσεις. Αν δε βρεθεί κανένα κοινό, το cell πετιέται και οι στατιστικοί πίνακες του user-network interface (UNI) αναβαθμίζονται. Ο δέκτης χρησιμοποιεί τον κώδικα καναλιού για να διαβάζει τις παραμέτρους καναλιού από τον receive connection table (RCT).

8.3.4 Ανασκόπηση του AAL5 Receiver

Ο δέκτης αντιγράφει το cell payload των 48 byte στον εξωτερικό ενταμιευτή και υπολογίζει το CRC-32 σε ολόκληρο το CPCS-PDU. Όταν έρθει το τελευταίο AAL5 cell ο δέκτης ελέγχει το μήκος, τα CRC-32, και CPCS-UU+CPI πεδία και θέτει τα αντίστοιχα RxBD status bits. Μπορεί να εκτελεστεί μια διακοπή για μία από τις 4 ουρές διακοπής. Ο δέκτης αντιγράφει το τελευταίο cell στη μνήμη, συμπεριλαμβάνοντας το padding και το AAL5 trailer. Το CPCS-UU+CPI (είσοδος 16 bit) μπορεί να διαβαστεί κατευθείαν από το AAL5 trailer.

Ο ελεγκτής ATM δείχνει τις καταστάσεις CLP και CNG των εισερχόμενων cell. Όταν κλείσει το μήνυμα αυτά τα ενδεχόμενα ορίζουν τα RxBD[CLP] και RxBD[CNG]. Όταν κανένας ενταμιευτής δεν είναι έτοιμος για να δεχτεί cell, ο δέκτης αλλάζει στην κατάσταση hunt και πετάει όλα τα πακέτα που είναι συσχετισμένα με το συγκεκριμένο frame (partial packet discard). Ο δέκτης προσπαθεί να ανοίξει νέους ενταμιευτές για αποδοχή cell μόνο μετά την άφιξη του τελευταίου cell των απορριπτόμενων AAL5 frame.

8.3.5 ATM Pace Control

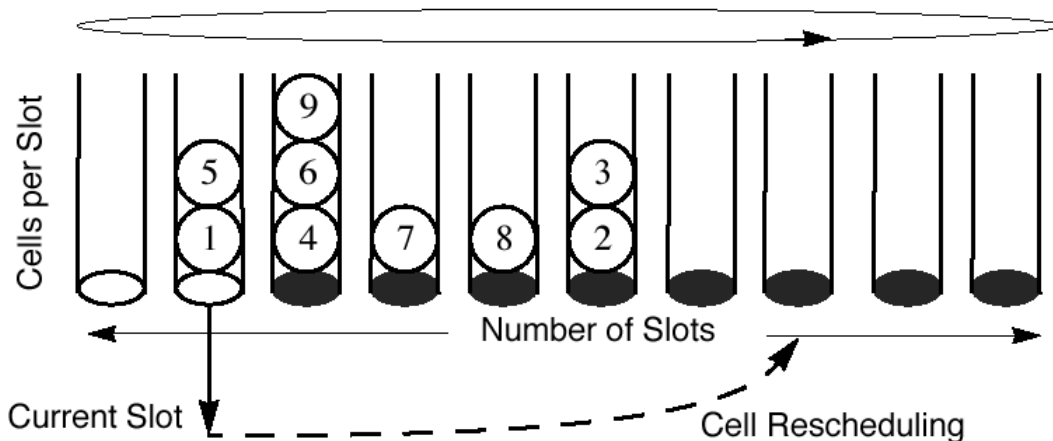
Η μονάδα ελέγχου βήματος του ATM (APC) προγραμματίζει τα κανάλια ATM για μετάδοση. Όσο εκτελείται αυτή η εργασία η μονάδα APC χρησιμοποιεί τις παρακάτω παραμέτρους:

- Συχνότητα (bandwidth) για κάθε κανάλι ATM.
- ATM traffic pacingNPeak cell rate (PCR), sustain cell rate (SCR), και minimum rate (MCR)
- Priority levelNReal-time channels (CBR or VBR-RT) προγραμματίζονται σε υψηλά επίπεδα προτεραιότητας. non-real-time channels (VBR-NRT, ABR, UBR) προγραμματίζονται σε χαμηλά επίπεδα προτεραιότητας. Είναι διαθέσιμα μέχρι και 8 επίπεδα προτεραιοτήτων.

Η μονάδα APC συνίσταται από μια δομή δεδομένων APC στη δίπορτη μνήμη RAM για κάθε ΡΗΥ και έναν ειδικό αλγόριθμο προγραμματισμού εκτελούμενο από το CP. Κάθε δομή δεδομένων ΡΗΥs APC περιλαμβάνει 3 στοιχεία: Έναν πίνακα παραμέτρων APC, έναν πίνακα προτεραιοτήτων APC και πίνακες προγραμματισμού μετάδοσης cell για κάθε επίπεδο προτεραιότητας.

Κάθε πίνακας παραμέτρων ΡΗΥ APC κρατάει παραμέτρους που καθορίζουν τη θέση του πίνακα προτεραιοτήτων, τον αριθμό των επιπέδων προτεραιοτήτων και άλλες παραμέτρους APC. Ο πίνακας προτεραιοτήτων κρατάει δείκτες που καθορίζουν τη θέση και το μέγεθος του πίνακα προγραμματισμού όλων των επιπέδων προτεραιοτήτων. Κάθε πίνακας προγραμματισμού που είναι διαιρεμένος σε time slots, όπως φαίνεται στην εικόνα 8.2.

Σχήμα 8.2: APC Scheduling Table Mechanism



Ο χρήστης καθορίζει τον αριθμό των ATM cells που αποστέλονται σε κάθε time slot (cells per slot). Μετά την αποστολή ενός καναλιού, αφαιρείται από το τρέχον time slot και προωθείται σε ένα μελλοντικό time slot σύμφωνα με το traffic rate που έχει ανατεθεί στο κανάλι (ορισμένο σε time slots). Η παράμετρος PCR στο TCT, ή οι παράμετροι SCR ή MCR στην επέκταση του TCT (TCTE) ορίζουν την πραγματική συχνότητα του καναλιού.

8.3.6 Address Lookup Mechanism

Το MPC8260 υποστηρίζει 2 τρόπους για το ψάξιμο διευθύνσεων για εισερχόμενα cell.

- Εξωτερική αναζήτηση στην CAM
- Συμπύεση διευθύνσεων

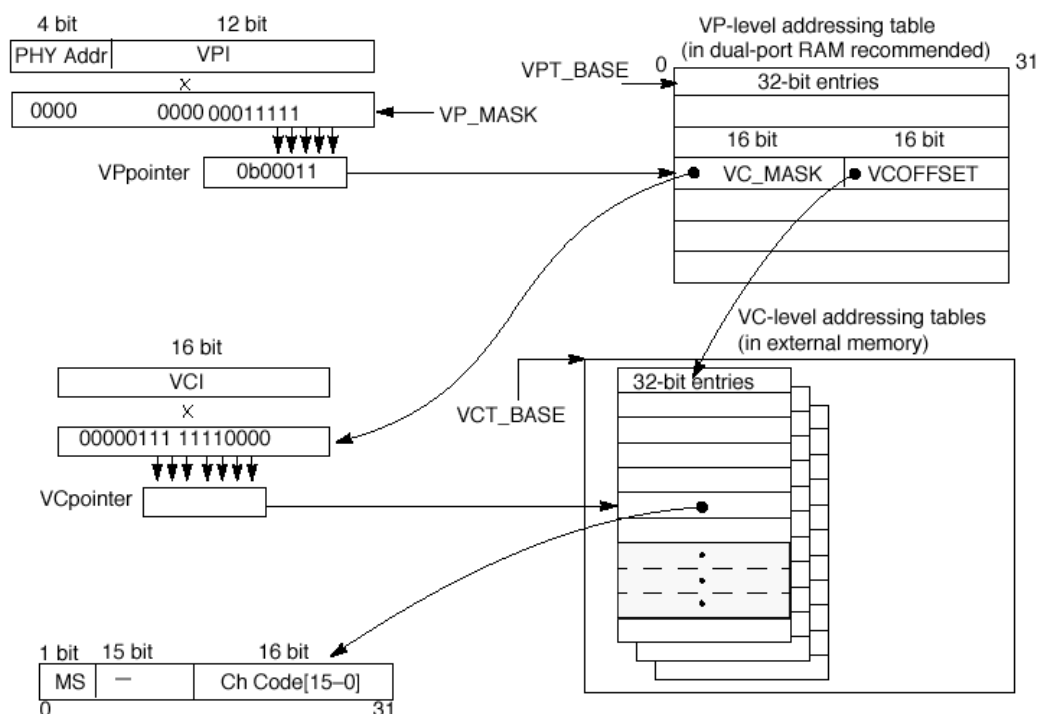
Ο μηχανισμός συμπίεσης διευθύνσεων χρησιμοποιεί 2 επίπεδα μετάφρασης διευθύνσεων για να βοηθήσει την ελαχιστοποίηση του χώρου μνήμης που απαιτείται για να καλυφθεί το διαθέσιμο πεδίο διευθύνσεων. Το πρώτο επίπεδο μετάφρασης (VP-level) χρησιμοποιεί έναν πίνακα αναζήτησης βασισμένο στην διεύθυνση 4 bit PHY και στον 12-bit virtual path identifier. Το δεύτερο επίπεδο μετάφρασης (VC-level) χρησιμοποιεί έναν 16-bit virtual channel identifier.

Εαν δεν έχουμε κάποια ταυτιση κατα την address compression, το πακέτο θεωρείται πως είναι misinserted. Κατα την μετάφραση, η VP_MASK μέσα στην ATM parameter RAM, συμπιέζει την φυσική διεύθυνση και το VPI από ένα εισερχόμενο πακέτο για να δημιουργήσει ένα δείκτη στο VP-level table. Το VP-level table, αποτελείται από μια μάσκα (VC_MASK) και έναν δείκτη σε ένα απο τα VC-level tables (VCOFFSET). Πρέπει να σημειώσουμε ότι το VP table, μπορεί και να βρίσκεται μέσα στην εσωτερική dual-port RAM του επεξεργαστή.

Κατά την μετάφραση VC-level, το VCI συμπιέζεται με την VC_MASK για να παράγει έναν δείκτη σε μία θέση στο VC-level table, που περιέχει τον κώδικα καναλιού για τα reserved πακέτα. Το VC table μπορεί να βρίσκεται και στην εξωτερική δυναμική μνήμη.

Η εικόνα 8.3 δείχνει ένα παράδειγμα συμπίεσης διευθύνσεων.

Σχήμα 8.3: Μηχανισμός Συμπίεσης Διευθύνσεων



Το σχήμα 8.3 δείχνει το VP_MASK να επιλέγει πέντε VPI bits για να βρει τον δείκτη για το VP-level table. Το περιεχόμενο του VP-level table περιέχει μία 16-bit mask (VC_MASK) και το VC-level table offset (VCOFFSET) για το επόμενο στάδιο του address mapping. Η VC_MASK επιλέγει τα 4-10 bits του VCI, το οποίο χρησιμοποιείται μαζί με τα VCT_BASE, VCOFFSET για να δείξει τον κώδικα του καναλιού για το εισερχόμενο πακέτο.

8.3.7 Δομή μνήμης του ATM

Η δομή μνήμης του ATM περιλαμβάνει την parameter RAM, τα connection tables, OAM performance monitoring tables, την δομή για τα δεδομένα APC, τα BD tables, τον AAL1 sequence number protection table και τον πίνακα για τα στατιστικά UNI.

Τα receive και transmit connection tables (RCT, TCT) αποθηκεύουν τις παραμέτρους από τις συνδέσεις που έχει αρχικοποιήσει ο host ύστερα από το setup της σύνδεσης. Οι παράμετροι αυτοί περιλαμβάνουν τον τύπο του AAL, τις παραμέτρους για την κίνηση της σύνδεσης, τις παραμέτρους για τα BDs, και μερικούς προσωρινούς υπολογισμούς που χρησιμοποιούνται κατά το segmentation και reassembly (SAR). Το transmit connection table extension (TCTE) υποστηρίζει και ειδικές συνδέσεις που χρησιμοποιούν τα ABR, VBR και UBR+ services. Κάθε πίνακας καταλαμβάνει χώρο 32-byte.

Κάθε κανάλι ATM, έχει έναν συγκεκριμένο κώδικα που χρησιμοποιεί για να βρει το δικό του entry στα connection tables. Το πρώτο table έχει κωδικό ένα, το δεύτερο δύο, και τα λοιπά. Κώδικες μέχρι 255, δείχνουν εσωτερικά κανάλια, ενώ κώδικες μεγαλύτεροι από 255 δείχνουν σε εξωτερικά κανάλια. Το κανάλι ένα, είναι δεσμευμένο από την raw cell queue και δεν μπορεί να χρησιμοποιηθεί για άλλο λόγο. Ο κώδικας του καναλιού χρησιμοποιείται για να ορίσει ένα συγκεκριμένο VC, όταν στέλνουμε την εντολή **ATM TRANSMIT**, όταν αρχικοποιούμε την εξωτερική CAM ή τους πίνακες του address compression και όταν ο CP στέλνει μία διακοπή στην ουρά με τις διακοπές.

Receive Connection Table

Κάθε Receive Connection table, έχει μια συγκεκριμένη μορφή. Εκεί βρίσκεται όλη η απαραίτητη πληροφορία για την υποδοχή δεδομένων από ένα συγκεκριμένο κανάλι. Στο σχήμα 8.4 μπορούμε την μορφή του πίνακα αυτού, και πως διαχωρίζεται.

Ενας παρόμοιος πίνακας χρησιμοποιείται για την μετάδοση των πακέτων ATM. Αυτός ο πίνακας καλείται Transmit Connection Table, και μπορούμε να τον δούμε στο σχήμα 8.5. Και οι δύο αυτοί πίνακες Receive, Transmit έχουν μια μικρή περιοχή που είναι συγκεκριμένη για κάθε πρωτόκολο. Αυτό σημαίνει πως οι πίνακες για το AAL5 δεν είναι ίδιοι με αυτούς του AAL0. Εάν χρησιμοποιούμε το AAL5 πρωτόκολο, τότε έχουμε τρεις επιλογές για το μικρό αυτό συγκεκριμένο πεδίο. Μπορούμε να χρησιμοποιήσουμε την δομή για το ABR, η για το

Σχήμα 8.4: Receive Connection Table

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Offset + 0x00	—	GBL	BO	—	DTB	BIB	—	BUFM	SEGF	ENDF	—	—	—	—	INTQ	
Offset + 0x02	—	INF	—										ABRF	AAL		
Offset + 0x04	RX Data Buffer Pointer (RXDBPTR)															
Offset + 0x06																
Offset + 0x08	Cell Time Stamp															
Offset + 0x0A																
Offset + 0x0C	RBD_Offset															
Offset + 0x0E	Protocol Specific															
Offset + 0x10																
Offset + 0x12																
Offset + 0x14																
Offset + 0x16																
Offset + 0x18																
Offset + 0x1A	MRBLR															
Offset + 0x1C	—	PMT						RBD_BASE								
Offset + 0x1E	RBD_BASE												—	PM		

UBR, ή ακόμα και για το CBR. Μιας και ανάλογα με την κίνηση, έχουμε και διαφορετικά στοιχεία να υπολογίσουμε, έτσι και εδώ οι δομές αυτές περιέχουν πολλά διαφορετικά πεδία ανα πρωτόκολλο.

APC Data Structure

Η δομή δεδομένων APC συνίσταται από 3 στοιχεία: Τους πίνακες παραμέτρων APC για τις συσκευές PHY, τον πίνακα προτεραιοτήτων APC, και του πίνακες προγραμματισμού APC. Όπως βλέπουμε στην εικόνα 8.6,

έχουμε μόνο έναν πίνακα παραμέτρων για κάθε φυσική συσκευή. Κάθε μία απ αυτές τις συσκευές, έχει 8 προτεραιότητες και κάθε προτεραιότητα έχει έναν πίνακα προγραμματισμού.

Κάθε πίνακας παραμέτρων APC του PHY κρατάει παραμέτρους που ορίζουν τη θέση του πίνακα προτεραιοτήτων, τον αριθμό των επιπέδων προτεραιότητας και άλλες παραμέτρους APC. Ο πίνακας βρίσκεται στη δυαδική πύλη RAM. Η παράμετρος APCP_BASE, δείχνει στη βασική διεύθυνση του πίνακα προτεραιοτήτων PHY.

Για πολλαπλά PHY η δομή του πίνακα διπλασιάζεται. Κάθε πίνακας βρίσκεται σε 32 byte μνήμης. Η αρχική διεύθυνση κάθε πίνακα παραμέτρων δίδεται από το $APCP_BASE + PHY(x) \times 32$. Να σημειωθεί όμως ότι στην κατάσταση slave με πολλαπλά PHY, ο πίνακας παραμέτρων πάντα βρίσκεται στο APCP_BASE ανεξάρτητα από τη διεύθυνση του PHY.

Κάθε πίνακας παραμέτρων APC του PHY κρατάει δείκτες στον πίνακα προγραμματισμού APC κάθε επιπέδου προτεραιότητας. Βρίσκεται στη δυαδική πύλη RAM. Ο πίνακας προτε-

Σχήμα 8.5: Τρανσμιτ δυνεστιον Ταβλε

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Offset + 0x00	—	GBL	BO	—	DTB	BIB	AVCF	—	ATT	CPUU	VCON	INTQ					
Offset + 0x02	—	INF	—									ABRF	AAL				
Offset + 0x04	Tx Data Buffer Pointer (TXDBPTR)																
Offset + 0x06																	
Offset + 0x08	TBD_CNT																
Offset + 0x0A	TBD_OFFSET																
Offset + 0x0C	Rate Remainder								PCR Fraction								
Offset + 0x0E	PCR																
Offset + 0x10	Protocol Specific																
Offset + 0x12																	
Offset + 0x14																	
Offset + 0x16	APC Linked Channel (APCLC)																
Offset + 0x18	ATM Cell Header (VPI,VCI,PTI,CLP)																
Offset + 0x1a																	
Offset + 0x1C	—	PMT								TBD_BASE							
Offset + 0x1E	TBD_BASE												BNM	STPT	IMK	PM	

ραιότητας μπορεί να κρατήσει μέχρι 8 επιπεδα προτεραιότητας.

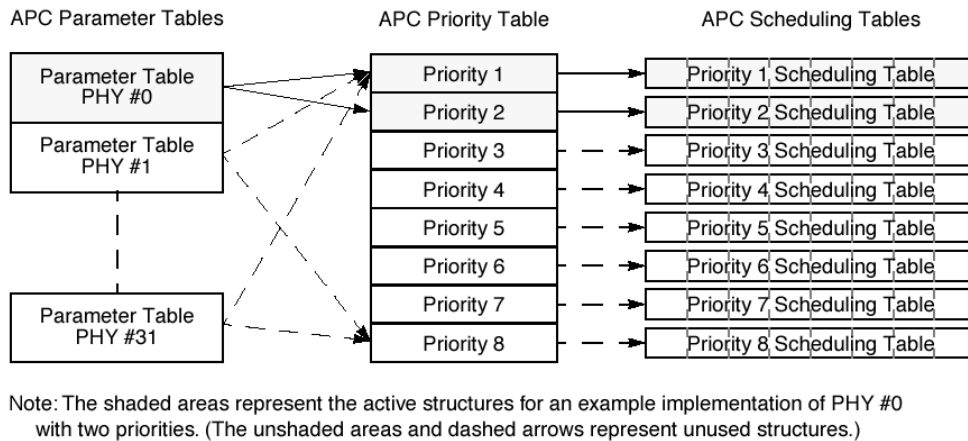
Το APC χρησιμοποιεί πίνακες προγραμματισμού APC (έναν για κάθε επιπεδο προτεραιότητας) για να προγραμματίσει μετάδοση καναλιών. Ένας πίνακας προγραμματισμού διαιρείται σε time slot. Κάθε slot είναι μια είσοδος μισής λέξης.

ATM Controller Buffer Descriptors (BDs)

Κάθε κανάλι ATM έχει ξεχωριστούς πίνακες BD αποδοχής και μετάδοσης. Ο αριθμός των BD ανά κανάλι και το μέγεθος των ενταμιευτών ορίζονται από το χρήστη. Το τελευταίο BD σε κάθε πίνακα κρατάει μια ένδειξη wrap. Κάθε BD στον πίνακα TxBD δείχνει σε έναν ενταμιευτή. Στην πλευρά του δέκτη, ο χρήστης μπορεί να διαλέξει μια από τις παρακάτω καταστάσεις.

- Στατική δέσμευση ενταμιευτή. Σε αυτήν την κατάσταση, ο χρήστης δεσμεύει dedicated buffers σε κάθε κανάλι ATM (δηλαδή ο χρήστης συσχετίζει κάθε BD με έναν ενταμιευτή). Η στατική δέσμευση ενταμιευτή είναι χρήσιμη όταν η συχνότητα σύνδεσης είναι γνωστή και σταθερή και όταν τα δεδομένα πρέπει να ξανασυγκεντρωθούν σε έναν συγκεκριμένο χώρο μνήμης.
- Ολική δέσμευση ενταμιευτή. Δυνατή μόνο για AAL5. Σ' αυτή την κατάσταση η δέσμευσή ενταμιευτή είναι δυναμική. Ο χρήστης δεσμεύει ενταμιευτές αποδοχής και τους τοποθετεί σε global buffer pools. Όταν το CP χρειάζεται έναν, πρώτα φέρνει έναν δείκτη ενταμιευτή από έναν από τους global buffer pools και γράφει το δείκτη στο τρέχον RxBD. Τα Global buffer allocation είναι βελτιστοποιημένα για να δεσμεύεται μνήμη

Σχήμα 8.6: ATM Pace Control Data Structure



ανάμεσα σε πολλά κανάλια ATM με διάφορες συχνότητες δεδομένων, όπως κανάλια ABR.

Ο χρήστης προετοιμάζει έναν πίνακα BD που δείχνει στους ενταμιευτές που θα αποσταλούν. Η διεύθυνση του πρώτου BD τοποθετείται στο TCT[TBD_BASE] του καναλιού. Η διαδικασία μετάδοσης αρχίζει όταν ο πυρήνας εκτελεί μια εντολή ATM TRANSMIT. Ο CP διαβάζει το πρώτο TxBD του πίνακα και στέλνει τον αντίστοιχο του ενταμιευτή. Όταν ο τρέχων ενταμιευτής τελειώσει, ο CP αυξάνει το TBD_Offset, που κρατάει το offset από το TBD_BASE στο τρέχον BD. Τότε διαβάζει το επόμενο BD του πίνακα. Αν το BD είναι έτοιμο (TxBD[R] = 1), ο CP συνεχίζει να στέλνει. Αν όχι, ο CP ελέγχει κατ' επανάληψη το έτοιμο bit στη συχνότητα καναλιού εκτός αν TCT[AVCF] = 1, οπότε ο CP αφαιρεί το κανάλι από το APC και διαγράφει το TCT[VCON]. Ο πυρήνας πρέπει να εκτελέσει μια νέα εντολή ATM TRANSMIT για να ξαναρχίσει η μετάδοση.

Η εικόνα 8.7 δείχνει το έτοιμο bit στους πίνακες TxBD και τους αντίστοιχους ενταμιευτές τους για δύο πρότυπα κανάλια ATM.

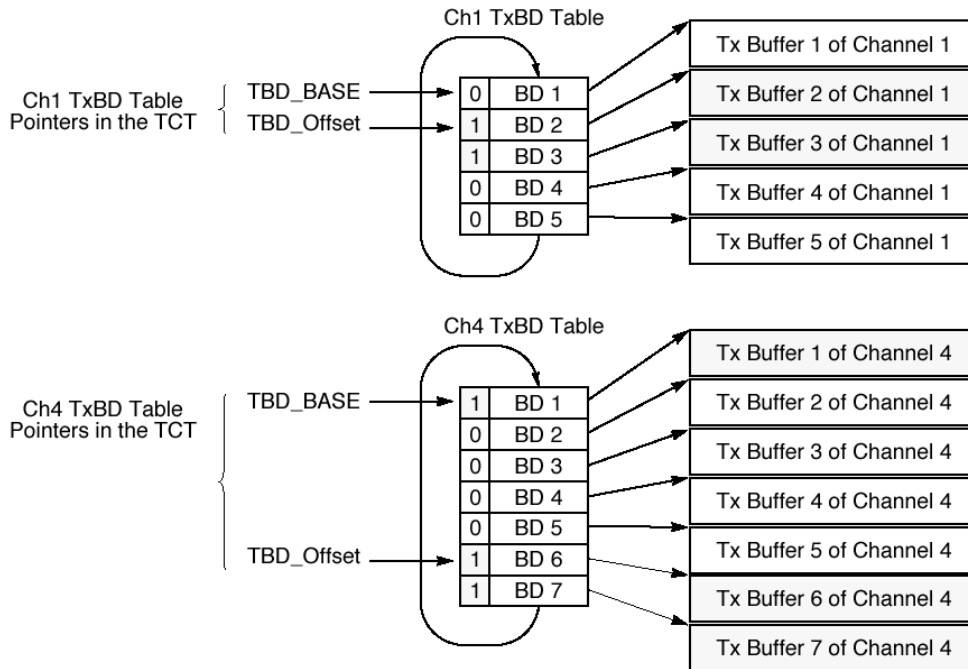
8.3.8 ATM Exceptions

Ο ελεγκτής του ATM που χειρίζεται διακοπές περιλαμβάνει 2 κύριες δομές δεδομένων: Τις FCCEs (FCC event registers) και τις κυκλικά διακοπόμενες ουρές.

Τέσσερις ουρές ιεραρχικής διακοπής είναι διαθέσιμες. Προγραμματίζοντας RCT[INTQ] και ο χρήστης καθορίζει ποιά ουρά θα δεχτεί τη διακοπή.

Μετά από μια αίτηση για διακοπή, ο host διαβάζει τον FCCE. Αν FCCE[GINTx] = 1 τουλάχιστον μία είσοδος έχει προστεθεί σε μια από τις ουρές. Αφού διαγραφεί το FCCE[GINTx], ο

Σχήμα 8.7: Ενταμιευτές Μετάδοσης και Παράδειγμα Πίνακα BD



host processes τις έγκυρες εισόδους των ουρών διακοπής και διαγράφει τα έγκυρα bit κάθε εισόδου. Ο host ακολουθεί αυτή τη διαδικασία μέχρι να βρει μια είσοδο με $V = 0$.

Ο host ελέγχει τον αριθμό των διακοπών που στέλνονται στον πυρήνα, χρησιμοποιώντας έναν μετρητή στον πίνακα παραμέτρων της ουράς διακοπής. Για κάθε ενδεχόμενο που στέλνεται σε μια ουρά, ένας μετρητής (που έχει αρχικοποιηθεί σε ένα threshold αριθμό διακοπών) μειώνεται. Όταν ο μετρητής φτάσει στο μηδέν, τίθεται η συνολική διακοπή (FCCE[GINTx]).

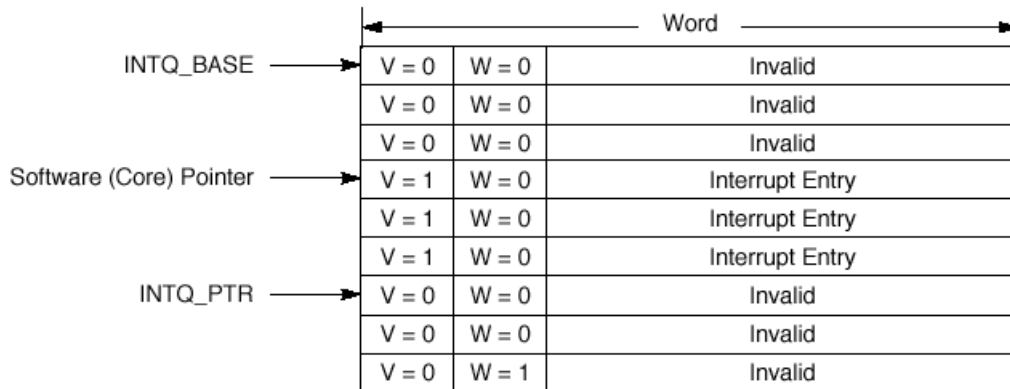
Interrupt Queues

Οι ουρές διακοπής βρίσκονται σε εξωτερική μνήμη. Οι παράμετροι κάθε ουράς αποθηκεύονται σε έναν πίνακα.

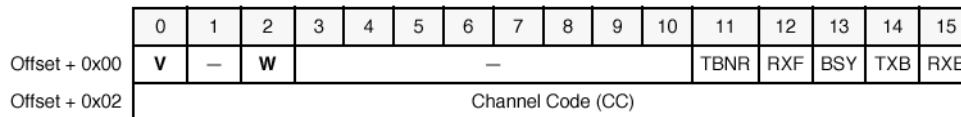
Όταν λαμβάνει χώρα μια διακοπή, το CP γράφει μια καινούργια είσοδο στην ουρά, ορίζεται το V bit, και αυξάνεται ο δείκτης της ουράς (INTQ_PTR). Τη στιγμή που το CP χρησιμοποιεί μια είσοδο με $W = 1$, επιστρέφει στην πρώτη είσοδο της ουράς. Αν το CP προσπαθήσει να κάνει everwrite μια έγκυρη είσοδο ($V = 1$) γίνεται μια κατάσταση υπερχείλισης και ορίζεται η σημαία υπερχείλισης της ουράς FCCE[INTOx]. Στην εικόνα 8.8 φαίνεται η δομή της ουράς διακοπής.

Στην εικόνα φαίνονται όλες οι εισοδοί της ουράς. Κάθε μονολεκτική είσοδος της ουράς παρέχει λεπτομερείς πληροφορίες για τη διακοπή στον host.

Σχήμα 8.8: Δομή Ουράς Διακοπής



Σχήμα 8.9: Δομή Ουράς Διακοπής



8.3.9 Utopia Interface

Ο ελεγκτής ATM επικοινωνεί με μια συσκευή PHY μέσω του UTOPIA interface. Ο MPC8260 υποστηρίζει UTOPIA επιπέδου 2 και για την κατάσταση master και για τη slave.

Το UTOPIA interface υποστηρίζει loop-back mode. Στην κατάσταση αυτή τα σήματα της Rx και Tx UTOPIA γίνονται shorted εσωτερικά internally. Τα output pins οδηγούνται, ενώ τα pins εισόδου αγνοούνται. Σημειώστε ότι στην κατάσταση loop-back ο μεταδότης και ο δέκτης πρέπει να λειτουργούν σε συμπληρωματικές καταστάσεις. Για παράδειγμα αν ο μεταδότης είναι master, ο δέκτης πρέπει να είναι slave (FPSMR[TUMS] = 0, FPSMR[RUMS] = 1). Οι καταστάσεις επιλέγονται μέσω του GFMR[DIAG].

8.4 ATM Linux Driver

8.4.1 Ανασκόπηση του Driver

Ο MPC8260 ATM driver, αποτελείται από δεκα αρχεία C, και δώδεκα αρχεία header.

Το αρχείο κώδικα Crpmtimer.c, περιέχει όλες τις ρουτίνες που χρειάζονται για την αρχικοποίηση και τον έλεγχο των χρονιστών και των baud rate generators. Το αρχείο Intpool.c,

Σχήμα 8.10: Driver's source and header files

C Source Files	Header Files
cpmtimer.c	cpm.h
intpool.c	cpmtimer.h
mm.c	debug.h
mpc8260sar.c	defines.h
mpc8260sar_detect.c	errata.h
mpool.c	intpool.h
risctimer.c	mm.h
idt77v106.c	mpc8260sar.h
vads_phy.c	mpool.h
utopia.c	risctimer.h
	utopia.h
	idt77v106.h

περιέχει μερικές ρουτίνες για τον έλεγχο και την αρχικοποίηση μιας μικρής integer pool. Αυτοί οι ακέραιοι χρειάζονται για την σωστή επιλογή των εσωτερικών τράνσμιτ και receive connection tables. Όταν δημιουργούμε ένα νέο VC, ψάχνουμε να βρούμε τον μικρότερο ακέραιο που είναι ελεύθερος, έτσι ώστε να επιλέξουμε και το πρώτο connection table που είναι ελεύθερο. Όταν ελευθερώνουμε ένα VC, ο ακέραιος που είχε δεσμευτεί, ελευθερώνεται, και τοποθετείται ξανά μέσα στην integer pool.

Το αρχείο κώδικα mm.c, περιέχει όλες τις απαραίτητες ρουτίνες για την διαχείριση της μνήμης. Μιας και ο CP και ο επεξεργαστής μοιράζονται το ίδιο δίαυλο και την ίδια μνήμη, το να έχουμε συνοχή είναι πολύ σημαντικό. Είναι εύκολο να διαπιστώσουμε τι καταστροφή θα γινόταν, εάν ο επεξεργαστής διάβαζε από την μνήμη παλιά δεδομένα, παρόλο που ο CPM θα έχει ανανεώσει τα δεδομένα μέσα στην μνήμη.

Το αρχείο mpc8260sar.c, είναι το κεντρικό και πιο σημαντικό αρχείο του οδηγητή. Περιέχει όλες τις απαραίτητες ρουτίνες για να ανοίξεις και να κλείνεις συνδέσεις, όπως επίσης και ρουτίνες για την αρχικοποίηση της μνήμης και διαχείρισης διακοπών. Αυτό είναι και το μεγαλύτερο αρχείο κώδικα του οδηγητή. Το αρχείο Mpc8260sar_detect.c, περιέχει την πρώτη συνάρτηση που καλείται μέσω του λειτουργικού συστήματος. Το mpool.c περιέχει διάφορες συναρτήσεις, για την διαχείριση της μνήμης του αρχείου που λαμβάνεται ή στέλνεται. Το αρχείο κώδικα utopia.c, περιέχει όλες τις απαραίτητες συναρτήσεις για την διαχείριση και αρχικοποίηση του utopia interface. Αυτό το αρχείο περιέχει ακόμα και μερικές συναρτήσεις για την ενεργοποίηση ή απενεργοποίηση των διακοπών του ATM interface.

Τέλος τα `idt77v106.c` και `vads_phy.c`, περιέχουν ορισμένες ρουτίνες για την αρχικοποίηση του PHY. Το πρώτο είναι για το πρωτότυπο InfoStar EP, ενώ το δεύτερο είναι για την πλατφόρμα αναπτυξης της Motorola, MPC8260 VADS.

Πρέπει να τονίσουμε πως ο οδηγητής ήταν γραμμένος αρχικά για τον επεξεργαστή MPC860. Αυτό που κάναμε, ήταν να μετατρέψουμε τον αρχικό κώδικα, έτσι ώστε να είναι συμβατός με το ATM interface του MPC8260. Μιας και οι διαφορές των δύο επεξεργαστών ήταν αρκετές, το μόνο που μπορούμε να πούμε πως έμεινε το ίδιο, είναι η δομή σχεδίασης του οδηγητή.

Όλα τα αρχεία header (.h), περιέχουν prototypes, καθώς και διαφορους ορισμούς σταθερών που χρειάζονταν μερικά αρχεία.

8.4.2 Η δομή του driver

Το αρχείο που καλείται πρώτα, είναι το αρχείο `mpc8260sar_detect.c`. Αυτό το αρχείο κώδικα, περιέχει την συνάρτηση `_init` η οποία καλείται από το λειτουργικό για να αρχικοποιήσει και να κάνει register τη νέα ATM συσκευή. Το registration γίνεται απο μία εξωτερική συνάσταση του συστήματος, την `atm_dev_register()`. Η αρχικοποίηση της συσκευής ATM, γίνεται από τη συνάρτηση `mpc8260sar_init`, μέσα στο αρχίο `mpc8260sar.c`.

Ψπάρχουν πολλά πράγματα που πρέπει να αρχικοποιηθούν από τον οδηγητή. Πρώτα πρέπει να αρχικοποιήσουμε μερικούς χρονιστές, όπως είναι ο `risc timer`. Αυτός ο χρονιστής, χρησιμοποιείται για να μαρκάρει τον χρόνο δημιουργίας των πακέτων. Δεύτερον, πρέπει να αρχικοποιήσει το `utopia interface`. Αυτό είναι αρκετά περίπλοκο. Έτσι καλούμε τη συνάρτηση `utopia_init`, μέσα από το αρχείο κώδικα, `utopia.c`. Η συνάρτηση αρχικοποίησης του `utopia`, είναι γραμμένη έτσι ώστε να υποστηρίζει τόσο το δικό μας πρωτότυπο, όσο και την αναπτυξιακή πλατφόρμα της Motorola. Οι βασικές διαφορές των δύο αυτών συστημάτων, είναι το `physical chipset`, και η αρχικοποίηση της πόρτας που είναι συνδεδεμένα αυτά.

Η συνάρτηση `mpc8260sar_init_dev_data` καλείται για να αρχικοποιήσει όλες τις απαραίτητες παραμέτρους για τις συνδέσεις ATM. Τα δεδομένα που αρχικοποιούνται, είναι ο αριθμός των BDs, ο αριθμός των VCI, VPI, και το `bitrate` του `physical`. Μετά έχουμε να υπολογίσουμε τις συγκεκριμένες παραμέτρους χρονισμού του `utopia`, καθώς επίσης και γενικές παραμέτρους χρονισμού του συστήματος. Έτσι καλούμε τις συναρτήσεις `utopia_calculate_timing` και `mpc8260sar_calculate_timing functions`.

Μέσα στο αρχείο κώδικα `utopia.c`, υπάρχουν και πολλές ακόμα κρίσιμες συναρτήσεις που καλούνται σχεδόν στην αρχή της διαδικασίας αρχικοποίησης. Μία απο αυτές αρχικοποιεί την πορτα με την οποία είναι συνδεδεμένο το `physical`, έτσι ώστε να έχουμε τις σωστές εξόδους και εισόδους. Άλλη μια πολύ σημαντική συνάρτηση, είναι η `utopia_enable_interrupts`. Αυτή η μικρή συνάρτηση κάνει register τη συγκεκριμένη διακοπή στο σύστημα και γράφει τις επιθυμητές στον καταχωρητή SIMRL, και κάνει enable τις διακοπές.

Αλλά δεν μπορούμε να έχουμε διακοπές, αν δεν μπορούμε να επικοινωνούμε με το εξωτερικό ολοκληρωμένο. Για να μπορούμε να έχουμε τη σωστή επικοινωνία, πρέπει να παρέχουμε το ρολόι στο ολοκληρωμένο αυτό. Για να το επιτύχουμε, πρέπει να αρχικοποιήσουμε έναν baud rate generator (BRG6). Όταν δοκιμάσαμε να το κάνουμε αυτό μέσω του λειτουργικού, μιας και είναι το πιο λογικό να κάνουμε, είχαμε κάποια σύγκρουση με την κάρτα δικτύου. Έτσι χρειάστηκε να βάλουμε αυτή την αρχικοποίηση πιο νωρίς, στο prcboot boot loader.

Το πρώτο σημάδι για το ότι πηγαίναμε προς τη σωστή κατεύθυνση, ήταν η ενεργοποίηση του led εισόδου του εξωτερικού ολοκληρωμένου. Όταν ενεργοποιήσαμε το baud rate generator, μπορούσαμε να δούμε το λαμπάκι να αναβοσβήνει κατά την λήψη των δεδομένων. Αλλά, παρ'όλα αυτά δεν είχαμε ακόμα διακοπές στο σύστημά μας.

Το πιο σημαντικό κομμάτι του οδηγητή, ήταν η συνάρτηση mpc8260sar_init_pram. Αυτή η συνάρτηση αρχικοποιούσε την εσωτερική dual port ram του επεξεργαστή, έτσι ώστε να μπορούμε να έχουμε υποστήριξη για ATM. Αυτή η συνάρτηση δεσμεύει το απαιτούμενο ποσό μνήμης μέσα στην dual port ram, ή στην εξωτερική δυναμική μνήμη, και αρχικοποιεί μερικές τιμές που πρέπει να γραφτούν σε συγκεκριμένες θέσεις μέσα σε αυτή. Για παράδειγμα αρχικοποιεί την FCC parameter RAM. Η FCC parameter RAM, έχει παρα πολλά πεδία που πρέπει να αρχικοποιηθούν. Μερικά από αυτά είναι η διεύθυνση των Ψέσεις και Transmit Connection Tables, την διεύθυνση του και τη διεύθυνση στην οποία θα αποθηκεύονται προσωρινά τα πακέτα καθώς και τη διεύθυνση του interrupt queue table.

Μέσα στη συνάρτηση init_pram, αρχικοποιούμε επίσης και την δομή των interrupt queues, καθώς επίσης και την διεύθυνση της δομής του address lookup mechanism. Ακόμα αρχικοποιεί τα scheduling tables και τις προτεραιότητες που χρειαζόμαστε. Τέλος υπολογίζει και αποθηκεύει δεδομένα για την κίνηση, όπως cells per slot και line rate.

Σε κάθε πεδίο πρέπει να τοποθετηθεί η σωστή τιμή με μεγάλη προσοχή. Δεν έχουμε την πολυτέλεια για ούτε ένα λάθος, μιας και ολόκληρος ο οδηγητής δεν πρόκειται να λειτουργήσει, και δεν θα μπορούμε εύκολα να βρούμε που είναι το λάθος. Μπορούμε να πούμε, πως η συγκεκριμένη λειτουργία είναι η καρδιά του οδηγητή. Αυτή η συνάρτηση αρχικοποιεί την δομή στην μνήμη που χρειάζεται ο επεξεργαστής για να λειτουργήσει.

Υστερα από την σωστή αρχικοποίηση της μνήμης, ο οδηγητής πρέπει να αρχικοποιήσει την integer pool και στη συνέχεια την raw cell queue. Αφού αρχικοποιήσουμε την integer pool, μετά πρέπει να αρχικοποιήσουμε το Receive Connection Table 0 για τα AALO packets. Η raw cell queue χρειάζεται για το AALO πρωτόκολλο. Μιας και τα management cells μεταδίδονται διαμέσου του AALO, πρέπει να την αρχικοποιήσουμε ένα θέλουμε OAM και RM cells. Εάν δεν αρχικοποιήσουμε την raw cell queue, όλα αυτά τα πακέτα χάνονται.

Τέλος, ο οδηγητής κάνει register την νέα συσκευή και ενεργοποιεί τις διακοπές καθώς επίσης και τον internal Baud Rate Generator. Τώρα όλα είναι έτοιμα για το User-Level interface.

8.4.3 Driver Interface

Ο πυρήνας του λειτουργικού συστήματος, χρειάζεται επιτά συναρτήσεις να γίνουν registered απο τον οδηγητή για το ATM. Αυτές οι συναρτήσεις είναι ίδιες για όλα τις συσκευές ATM στον πυρήνα linux 2.4.x. Οι συναρτήσεις αυτές, είναι οι ακόλουθες :

```
static int mpc8260sar_open(struct atm_vcc *vcc,short vpi,int vci);  
static void mpc8260sar_close(struct atm_vcc *vcc);  
static int mpc8260sar_ioctl(struct atm_dev *dev, unsigned int cmd,void *arg);  
static int mpc8260sar_setsockopt(struct atm_vcc *vcc,int level,int optname, void *optval,int optlen);  
static int mpc8260sar_getsockopt(struct atm_vcc *vcc,int level,int optname, void *optval,int optlen);  
static int mpc8260sar_send(struct atm_vcc *vcc,struct sk_buff *skb);  
static int mpc8260sar_sg_send(struct atm_vcc *vcc,unsigned long start, unsigned long size);  
static void mpc8260sar_phy_put(struct atm_dev *dev,unsigned char value, unsigned long addr);  
static unsigned char mpc8260sar_phy_get(struct atm_dev *dev,unsigned long addr);  
static int mpc8260sar_change_qos(struct atm_vcc *vcc,struct atm_qos *qos,int flgs);  
static int mpc8260sar_proc_read(struct atm_dev *dev,loff_t *pos,char *page);
```

Η συνάρτηση **mpc8260sar_open()** ανοίγει μία νέα σύνδεση σε ένα συγκεκριμένο VPI,VCI, με ένα συγκεκριμένο QoS. Η πληροφορία για το QoS, μαζί με μερικές άλλες παραμέτρους, βρίσκονται στην δομή vcc. Η συνάρτηση **mpc8260sar_close()**, κάνει ακριβώς το αντίθετο. Κλείνει μία ανοικτή σύνδεση. Στον οδηγητή μας, οι συναρτήσεις **mpc8260sar_ioctl()**, **mpc8260sar_setsockopt()**,**mpc8260sar_getsockopt()**, **mpc8260sar_sg_send()**, δεν έχουν υλοποιηθεί. Απλώς επιστρέφουν λάθος, αν καλεστούν.

Η συνάρτηση **mpc8260sar_send()** είναι αυτή που στέλνει τον ενταμιευτή με τα δεδομένα που θέλουμε να μεταφέρουμε. Στη συνέχεια η συνάρτηση **mpc8260sar_phy_put()** γράφει έναν 8 bit αριθμό στο εξωτερικό ολοκληρωμένο. Με την συνάρτηση αυτή, ο χρήστης μπορεί να θέσει το physical σε loopback mode. Η συνάρτηση **mpc8260sar_phy_get()** δεν έχει υλοποιηθεί στον οδηγητή μας, μιας και είχαμε ένα πρόβλημα με την επικοινωνία με το εξωτερικό ολοκληρωμένο. Η επόμενη συνάρτηση είναι κοινή για κάθε οδηγητή, και αλλάζει την πληροφορία για το QoS. Τέλος, η συνάρτηση **mpc8260sar_proc_read()** χρησιμοποιείται για να παρέχει χρήσιμες πληροφορίες στο χρήστη μέσω του proc filesystem (πχ, τον αριθμό των πακέτων που στάλθηκαν).

8.5 Physical Custom Driver

Στο InfoStar EP, έχουμε χρησιμοποιήσει το νέο ολοκληρωμένο 3.3V, IDT77V106. Αυτό το ολοκληρωμένο έχει ένα μικρό δίαυλο για να ελέγχεται. Όλα τα σήματα είναι συνδεδεμένα σε GPIO pins. Αυτοί οι καταχωρητές γίνανε memory mapped, για να μπορεί το λειτουργικό να τα προσπελεύσει σε συγκεκριμένη διεύθυνση.

Το Utility Bus είναι ένα byte-wide interface που παρέχει πρόσβαση στους καταχωρητές του IDT77V106. Αυτοί οι καταχωρητές χρησιμοποιούνται για να διαλέξουμε τα κατάλληλα χαρακτηριστικά λειτουργίας και συναρτήσεις, και για να βλέπουμε την κατάσταση των άλλων συστημάτων. Ο δίαυλος αυτός, είναι υλοποιημένος χρησιμοποιώντας πολυπλεγμένα τα δεδομένα με τις διευθύνσεις (AD[7:0]) όπου ο καταχωρητής διευθύνσεων ενεργοποιείται με το σήμα Address Latch Enable (ALE).

Το Utility Bus interface αποτελείται από τα ακόλουθα pins:

AD[7:0], ALE, CS, RD, WR

Στο σχήμα 8.11, μπορούμε να δούμε τις κυματομορφές για την λειτουργία της εγγραφής και της ανάγνωσης.

Read Operation

A register read is performed as follows:

1. Initial condition: RD, WR, CS not asserted (logic 1) ALE not asserted (logic 0)
2. Set up register address: place desired register address on AD[7:0] set ALE to logic 1; latch this address by setting ALE to logic 0.
3. Read register data: Remove register address data from AD[7:0] assert CS by setting to logic 0; assert RD by setting to logic 0 wait minimum pulse width time (see AC specifications)

Write Operation

A register write is performed as described below:

1. Initial condition: RD, WR, CS not asserted (logic 1) ALE not asserted (logic 0)
2. Set up register address: place desired register address on AD[7:0] set ALE to logic 1; latch this address by setting ALE to logic 0.

Σχήμα 8.11: Κυματομορφές χρονισμού του Utility Bus

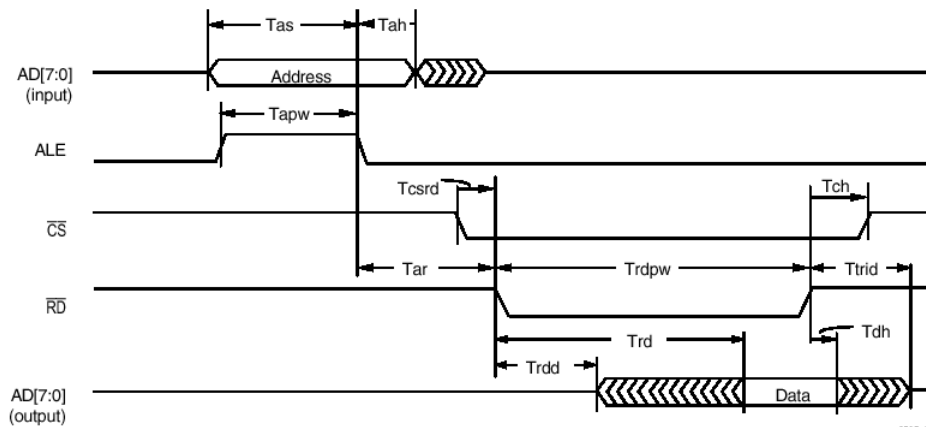


Figure 19 Utility Bus Read Cycle

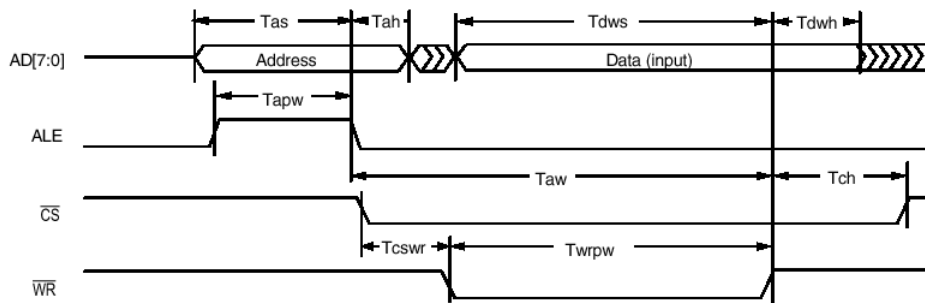


Figure 20 Utility Bus Write Cycle

- Write data: place data on AD[7:0] assert CS by setting to logic 0; assert WR (logic 0) for minimum time (according to timing specification); reset WR or CS to logic 1 to complete register write cycle.

Παρόλο που έμοιαζε αρκετά ευκολο, δεν ήταν. Λόγο προβλήματος στην σχεδίαση του τυπομένου, δεν μπορούσαμε να αρχικοποιήσουμε το εξωτερικό ολοκληρωμένο. Στην πραγματικότητα δεν μπορούσαμε να οδηγήσουμε τα σήματα.

Για να μπορέσουμε να υπερβούμε το πρόβλημα, επιλέξαμε να προγραμματίσουμε τις πόρτες GPIO μόνο σαν εξόδους. Έτσι μπορέσαμε να γράψουμε στο ολοκληρωμένο ότι θέλαμε. Αυτό ήταν πολύ σημαντικό, μιας και το ολοκληρωμένο περιείχε πολλές επιλογές που έπρεπε να ενεργοποιήσουμε. Για παράδειγμα θέλαμε το loopback mode. Μπορούσαμε να επιλέξουμε την λειτουργία αυτή, έτσι ώστε να δοκιμάσουμε το αναλογικό κομμάτι του ATM.

Αλλά δεν γνωρίζαμε αν ο οδηγητής μπορούσε να γράψει κάτι στο ολοκληρωμένο. Έπρεπε πρώτα να διαβάσουμε τους εσωτερικούς τους καταχωρητές πρώτα, για να είμαστε σίγουροι. Για να το καταφέρουμε αυτό, έπρεπε να αλλάξουμε τις ρυθμίσεις της πόρτας από έξοδο σε είσοδο, και αντίστροφα. Αναλόγως την λειτουργία που θέλαμε, ρυθμίζαμε και την πορτα να

είναι αντίστοιχα έξοδος ή είσοδος.

Έτσι, μιας και δεν μας ενδιαφέρει η ταχύτητα, μπορέσαμε να γράψουμε και να διαβάσουμε τους καταχωρητές του εξωτερικού ολοκληρωμένου. Αλλά, αυτή η συνεχής αλλαγή της κατάστασης της πόρτας, επιρρέαζε την κάρτα δικτύου. Αυτό σημαίνει πως έπρεπε να αρχικοποιήσουμε την πόρτα μόνο μία φορά. Έτσι, μιας και η διαδικασία ανάγνωσης δεν ήταν απαραίτητη, την αφαιρέσαμε από τον οδηγητή. Ο αριθμός των γραμμών των δύο αυτών αρχείων (idt77v106.c and idt77v106.c), είναι 304.

Συμπεράσματα

Τα συν και τα πλήν των κεντρικοποιημένων home gateways, κάπως μοιάζουν με αυτά των ενσωματωμένων, σε σχέση με βασισμένα σε κομμάτια, στερεοφωνικά συστήματα.

Απο την μία μεριά, ένα ενσωματωμένο στερεοφωνικό σύστημα, προσφέρει μια πιο έξυπνη, πιο φορητή, με λογική ποιότητα ήχου λύση. Από την άλλη όμως, τα ενσωματωμένα συστήματα μπορεί να είναι πιο δύσκολα να διορθωθούν, και δεν σου παρέχουν την δυνατότητα να επιλέξεις μερικά χαρακτηριστικά σε συγκεκριμένα τμήματά του. Το πιο πιθανό είναι, πως τα centralized gateways θα βρούμε θέση σε μερικά δίκτυα σπιτιών, αλλά δεν θα γίνουν ποτέ η υπέρτατη δικτυακή λύση για το σπίτι.

Το InfoGate project, ήταν ένα απο τα πρώτα βήματα στην αγορά των Residential Gateway. Αλλά η αγορά είναι απαιτητική, και ζητάει όλο και περισσότερα σε χαμηλότερη τιμή και σε μικρότερο μέγεθος. Το InfoGate Compact ήρθε για να γεμίσει το κενό αυτό στην αγορά.

Αλλά τίποτα δεν είναι δεδομένο. Κάθε μέρα που περνάει, ο καταναλωτής ζητάει όλο και περισσότερα. Άρα η μόνη επιλογή είναι η συνεχής ενημέρωση και βελτίωση της τεχνολογίας. Και μιας και τα Residential Gateways είναι ακόμα ανώριμα, χρειάζεται πολύ δουλειά για ένα επιτυχημένο RG προϊόν.

Βιβλιογραφία

- [1] Accel Technologies. PCAD 2001 Accel Technologies.
- [2] Cadance SoftWare. Cadence Specctra Software for Routing.
- [3] Efficient Networks,Inc. Residential Gateway: the Hub of a Broadband Home.
- [4] Holliday, Clifton. Residential Gateway.
- [5] Holliday, Clifton. The Residential Gateway Functional Specification.
- [6] Mark Anderson Karl. The Residential Gateway: Expanding the Horizons of Home Networking.
- [7] Martin Aragon and Susan Romano and James Tummins and Maria Stachelek. Residential Gateway Viability.
- [8] Motorola Semiconductors. MPC8260 User Manual, MPC8260 Errata.
- [9] Schoechle, Timothy, Conversation with author. *Boulder, Colorado, November 4. 1998.*
- [10] Unknown. Java Joins Home Connectivity Craze, Home Networking News.