

Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

Υλοποίηση ενός υποσυστήματος Αντιστάθμισης Πληροφορίας Κίνησης για έναν αποκωδικοποιητή MPEG-II

Σπύρος Γ. Λυμπέρης
Μεταπτυχιακή Εργασία

Ηράκλειο, Ιούνιος 2002

Μεταπτυχιακή εργασία
που υποβλήθηκε στην επιτροπή μεταπτυχιακών σπουδών
του Τμήματος Επιστήμης Υπολογιστών
της Σχολής Θετικών Επιστημών
του Πανεπιστημίου Κρήτης
σε μερική εκπλήρωση των απαιτήσεων για την απόκτηση του
Διπλώματος Μεταπτυχιακής Ειδίκευσης
Ηράκλειο, Ιούνιος 2002

Συγγραφέας:

Σπύρος Λυμπέρης
(Τμήμα Επιστήμης Υπολογιστών)

Εξετ. Επιτροπή:

Μανόλης Κατεβαίνης, Καθηγητής
(Εισηγητής)

Απόστολος Τραγανίτης, Αναπλ. Καθηγητής
(Μέλος)

Ευάγγελος Μαρκάτος, Αναπλ. Καθηγητής
(Μέλος)

Χρήστος Σωτηρίου, Επισκέπτης Καθηγητής
(Μέλος)

Δεκτή:

Πάνος Κωνσταντόπουλος, Καθηγητής
(Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών)

Υλοποίηση ενός υποσυστήματος Αντιστάθμισης Πληροφορίας Κίνησης για έναν αποκωδικοποιητή MPEG-II

Περίληψη

Στην εργασία αυτή περιγράφεται η ανάπτυξη ενός υποσυστήματος για ένα ολοκληρωμένο κύκλωμα. Το υποσύστημα αυτό είναι το ένα από τα δύο βασικά μέρη ενός αποκωδικοποιητή MPEG-II και ασχολείται με την δημιουργία των εικόνων πρόβλεψης του ρεύματος video. Ο σχεδιασμός του ξεκίνησε από το αρχικό στάδιο της μελέτης των απαιτήσεων και κατέληξε στην προδιαγραφή των μασκών κατασκευής για ένα πυρήνα πυριτίου. Στο κείμενο περιέχονται με λεπτομέρεια η μελέτη και ο σχεδιασμός του κυκλώματος, τα εργαλεία που χρησιμοποιήθηκαν, η μεθοδολογία που ακολουθήθηκε και οι επιδόσεις του τελικού συστήματος. Τέλος, γίνεται μια αποτίμηση της εμπειρίας που αποκτήθηκε κατά την εκπόνηση της εργασίας και αξιολογείται η δυσκολία των διάφορων σταδίων της.

Επόπτης: Μανόλης Κατεβαίνης

Implementation of a Motion Compensation Subsystem for an MPEG-II decoder

Summary

In this thesis an implementation of an Integrated Circuit subsystem is being described. This subsystem is one of the two major parts of an MPEG-II decoder which is responsible for the creation of predictor images of the video stream. Its design started from the initial stage of the block requirements and ended on the mask layout of the IC Core. In this document there are details on the study and the design of the circuit, the tools that were used, the methodology that was followed and the performance of the resulting block. Finally, an evaluation of the acquired experience during this thesis takes place and the difficulty of the various stages is explained.

Supervisor: Manolis Katevenis

Περιεχόμενα

| | |
|---|-----------|
| Περίληψη | i |
| Summary | i |
| Περιεχόμενα | iii |
| Κατάλογος Σχημάτων | v |
| Κατάλογος Πινάκων | vii |
| 1 Εισαγωγή | 1 |
| 1.1 Οπτικοακουστική πληροφορία και κωδικοποίηση | 1 |
| 1.2 Υλοποιήσεις σε Hardware | 2 |
| 1.3 Περιεχόμενο αυτής της εργασίας | 3 |
| 1.4 Οργάνωση του παρόντος κειμένου | 4 |
| 2 Το σύστημα κωδικοποίησης MPEG-II | 5 |
| 2.1 Βασική αρχή | 5 |
| 2.2 Η δομή ενός ρεύματος MPEG-II | 8 |
| 2.3 Η δομή ενός Στοιχειώδους Ρεύματος Video | 9 |
| 2.4 Macroblocks και κωδικοποίηση 4:2:0 | 10 |
| 3 Το υποσύστημα ΑΠΚ | 13 |
| 3.1 Περιβάλλον του υποσυστήματος | 13 |
| 3.2 Τύποι εικόνων και προβλέψεων | 14 |

| | | |
|----------|---|-----------|
| 3.2.1 | Αριθμός εικόνων πηγής | 14 |
| 3.2.2 | Τρόποι πρόβλεψης | 15 |
| 3.2.3 | Φίλτρο μισού pixel | 16 |
| 3.3 | Επεξεργασία διανυσμάτων | 17 |
| 3.4 | Μορφή εξόδου προς το μίκτη | 18 |
| 3.5 | Ρολόγια | 18 |
| 4 | Μελέτη και σχεδίαση | 21 |
| 4.1 | Ανάλυση απαιτήσεων | 21 |
| 4.2 | Καταμερισμός σε μπλοκ | 22 |
| 4.2.1 | Μπλοκ Εισόδου Διανυσμάτων | 23 |
| 4.2.2 | Μπλοκ Αιτήσεων Μνήμης | 23 |
| 4.2.3 | Μπλοκ Εισόδου Παραθύρων | 23 |
| 4.2.4 | Μπλοκ Μονάδας Κυρίως Επεξεργασίας | 24 |
| 4.2.5 | Μπλοκ Αναπροσαρμοστικής Εξόδου | 26 |
| 4.3 | Συγχρονισμός | 26 |
| 5 | Μεθοδολογία και εργαλεία | 29 |
| 5.1 | Ανάλυση και προσχέδιο | 29 |
| 5.1.1 | Κείμενο Βασικών Απαιτήσεων | 29 |
| 5.1.2 | Κείμενο Περιγραφής Σχεδίασης | 30 |
| 5.1.3 | Προβλήματα και Διλήμματα | 31 |
| 5.2 | Περιγραφή σε VHDL | 32 |
| 5.3 | Έλεγχος και ανάπτυξη εργαλείου | 32 |
| 5.4 | Σύνθεση στα 0.18 μm και Χρονική Ανάλυση | 33 |
| 5.5 | Τοποθέτηση και διασύνδεση | 35 |

| | |
|----------------------------------|-----------|
| 6 Αποτελέσματα | 37 |
| 7 Επίλογος | 41 |
| 7.1 Εμπειρία | 41 |
| 7.2 Μελλοντική εργασία | 42 |
| 8 Αναφορές | 43 |

Κατάλογος Σχημάτων

| | | |
|-----|--|----|
| 2.1 | Απλή μετάδοση video | 6 |
| 2.2 | Πρόβλεψη στάσιμης περιοχής | 7 |
| 2.3 | Πρόβλεψη κινούμενης περιοχής | 8 |
| 2.4 | Σύστημα χρωμάτων 4:2:0 | 11 |
| 3.1 | Περιβάλλον του ΑΠΚ | 14 |
| 3.2 | Αριθμός εικόνων πηγής | 15 |
| 3.3 | Φίλτρο μισού pixel | 17 |
| 3.4 | Ρολόγια του ΑΠΚ | 18 |
| 4.1 | Χρονική ροή των macroblocks | 22 |
| 4.2 | Μπλοκς του ΑΠΚ | 22 |
| 5.1 | Δομή κειμένων του υποσυστήματος | 31 |
| 6.1 | Τοποθέτηση και διασύνδεση ενός μπλοκ | 40 |

Κατάλογος Πινάκων

| | | |
|-----|--|----|
| 1.1 | Ενδεικτικές τιμές εύρους ζώνης για video | 2 |
| 2.1 | Είδη εικόνων σε ένα Video ES | 10 |
| 6.1 | Στοιχεία του ΑΠΚ | 38 |
| 6.2 | Χρόνος για την ανάπτυξη του ΑΠΚ | 38 |
| 6.3 | Εργαλεία που χρησιμοποιήθηκαν | 39 |

1

Εισαγωγή

1.1 Οπτικοακουστική πληροφορία και κωδικοποίηση

Στις μέρες μας οι εφαρμογές που χρησιμοποιούν οπτικοακουστική πληροφορία (video) εξαπλώνονται ολοένα και περισσότερο. Η γνωστή μας αναλογική τηλεόραση και το – συνήθως συνοδευτικό – οικιακό βιντεοκασσετόφωνο τείνουν προς την πλήρη τους ψηφιοποίηση. Ήδη, η ψηφιακές πλατφόρμες για δορυφορική τηλεόραση (DVB-S) και ψηφιακή αναπαραγωγή ταινιών (DVD Video) έχουν εδραιωθεί. Ταυτόχρονα, στο επίπεδο των προσωπικών υπολογιστών, κανείς διαπιστώνει την ευρεία εξάπλωση της πληροφορίας video σε παιχνίδια, σε ιστοσελίδες ή ακόμα και σε μέχρι πρότινος κοινές εφαρμογές γραφείου.

Αυτή η εξέλιξη δεν θα ήταν ποτέ δυνατή χωρίς τη χρήση κωδικοποίησης για την οπτικοακουστική πληροφορία. Το πολύ μεγάλο για τα σημερινά δεδομένα εύρος ζώνης που απαιτείται για την μετάδοση μη κωδικοποιημένου σήματος video το καθιστά απαγορευτικό προς διακίνηση στους υπάρχοντες δίαυλους. Στον πίνακα 1.1 φαίνονται μερικές ενδεικτικές τιμές για απλή μετάδοση εικόνων σε σχέση με τη χρήση ενός συστήματος κωδικοποίησης MPEG-II.

Η κωδικοποίηση της πληροφορίας γίνεται με μια σειρά τεχνικών, που βασίζονται αφ' ενός

| Μέγεθος εικόνας | Εύρος (κανονικό) | Εύρος (MPEG-II) |
|-----------------|------------------|-----------------|
| 1920 x 1152 | 638 Mbps | 80 Mbps |
| 720 x 576 | 120 Mbps | 15 Mbps |
| 352 x 288 | 30 Mbps | 4 Mbps |

Πίνακας 1.1: Ενδεικτικές τιμές εύρους ζώνης για video

στη συμπίεσή της και αφ' ετέρου στην "παρατήρηση" ομοιοτήτων ανάμεσα στις εικόνες. Μια εισαγωγή σε τέτοιες λεπτομέρειες θα γίνει στο κεφάλαιο 2.

1.2 Υλοποιήσεις σε Hardware

Το κατά πόσον χρειάζεται να υλοποιούνται κωδικοποιητές ή αποκωδικοποιητές video σε hardware, εξαρτάται πάντα από τις ανάγκες της εκάστοτε εφαρμογής. Λόγω του μεγέθους και των δυσκολιών των μεθόδων που χρησιμοποιούνται σήμερα, υπάρχουν υλοποιήσεις και σε hardware αλλά και σε software.

Ειδικότερα για το MPEG-II, ο κωδικοποιητής καλείται να συμμετέχει σε μια υπερβολικά χρονοβόρα και ενεργοβόρα διαδικασία για τη συμπίεση των δεδομένων. Την ίδια στιγμή όμως, κωδικοποιητές που να χρησιμοποιούνται σε πραγματικό χρόνο δεν είναι διαδεδομένοι. Συνήθως λοιπόν, υλοποιήσεις σε hardware τέτοιων αλγορίθμων συναντούνται κυρίως σε επαγγελματικούς χώρους (στούντιο κινηματογράφων ή ψηφιακών τηλεοπτικών σταθμών). Υπάρχουν πολλές εφαρμογές κωδικοποίησης MPEG-II που γίνονται από λογισμικό σε γρήγορους επεξεργαστές γενικού σκοπού, που αφήνονται για όσο χρόνο είναι απαραίτητο.

Απο την άλλη μεριά, οι αποκωδικοποιητές video είναι σχεδόν πάντα συνυφασμένοι με την απόδοση σε πραγματικό χρόνο: ο χρήστης βλέπει το αποκωδικοποιημένο σήμα στην τηλεόραση ή στον υπολογιστή του. Σήμερα οι υπολογιστές γενικού σκοπού δεν καταφέρνουν να αποκωδικοποιήσουν σήματα MPEG-II απλής ευκρίνειας (τηλεόρασης) με απόδοση 100%. Αντίθετα, τα συστήματα που χρησιμοποιούνται είναι κυρίως δύο: πλήρους αποκωδικοποίησης σε hardware και μερικής αποκωδικοποίησης σε hardware.

Στα πρώτα, ένα σύστημα σε hardware αναλαμβάνει πλήρως την αποκωδικοποίηση ξεκινώντας από το μεταδιδόμενο σήμα (συνήθως σε ψηφιακή μορφή) και καταλήγοντας σε απλό σήμα τηλεόρασης ή οθόνης υπολογιστή. Στα δεύτερα, ένα ολοκληρωμένο κύκλωμα που βρίσκεται σε κάποια κάρτα μέσα σε έναν υπολογιστή υποβοηθάει το λογισμικό στις "βαριές" υπολογιστικές πτυχές της αποκωδικοποίησης. Και οι δύο αυτές μορφές είναι σε ευρεία χρήση. Οι συσκευές DVD Players και οι κάρτες επέκτασης πλήρους αποκωδικοποι-

σης είναι οι πιο προφανείς εφαρμογές της πρώτης κατηγορίας. Τα ολοκληρωμένα της δεύτερης κατηγορίας συνήθως βρίσκονται πάνω σε κάρτες γραφικών προσωπικών υπολογιστών, ως “συμπλήρωμα” της λειτουργικότητάς τους.

1.3 Περιεχόμενο αυτής της εργασίας

Η υλοποίηση του υποσυστήματος Αντιστάθμισης Πληροφορίας Κίνησης που παρουσιάζεται εδώ, ανήκει σε ένα γενικότερο σύστημα πλήρους αποκωδικοποίησης για οπτικοακουστικό σήμα MPEG-II σε συνηθισμένη ευκρίνεια (μέγεθος εικόνας 720 x 576). Το όλο σύστημα είναι σχεδιασμένο για την αναπαραγωγή μέχρι και δύο ταυτόχρονων τέτοιων σημάτων.

Η σχεδίαση του συστήματος είναι βασισμένη σε ένα διεθνές πρότυπο της IEEE ([1]), που ορίζει επακριβώς τα σήματα και τους αλγόριθμους για τη μετάδοση και την επεξεργασία video σε μορφή MPEG-II. Το σύστημα είναι σχεδιασμένο απολύτως σύμφωνα με αυτές τις οδηγίες, στοχεύοντας περισσότερο σε μια ανταγωνιστική έκδοση ενός πυρήνα σε ολοκληρωμένο κύκλωμα (chip core) σε λίγα τετραγωνικά χιλιοστά.

Το συγκεκριμένο υποσύστημα που υλοποιήθηκε είναι το ένα από τα δύο βασικά υπολογιστικά μέρη του αποκωδικοποιητή. Όπως θα εξηγηθεί στο κεφάλαιο 2, το υποσύστημα αυτό χειρίζεται τα διανύσματα της κίνησης που αντλούν πληροφορία από προηγούμενες και επόμενες εικόνες. Στο κείμενο αυτό περιγράφεται αυτό το κύκλωμα και εξηγείται η μεθοδολογία της υλοποίησής του, από το αρχικό στάδιο της περιγραφής της λειτουργίας του μέχρι και την κατασκευή της περιγραφής των μασκών του πυριτίου. Αναλυτικότερα, η εργασία αποτελείται από τα εξής κομμάτια:

- ⇒ **Κείμενα Βασικών Απαιτήσεων και Περιγραφής Σχεδίασης:** καθορισμός των απαιτήσεων και σχεδίαση του υποσυστήματος. Καταγραφή των σταδίων σε κείμενα.
- ⇒ **Ανάπτυξη κώδικα VHDL και εργαλείου ελέγχου:** υλοποίηση στη γλώσσα περιγραφής hardware VHDL και ανάπτυξη ενός ειδικού εργαλείου που βοηθάει στον τελικό έλεγχο.
- ⇒ **Έλεγχος:** επιβεβαίωση της σωστής λειτουργίας του κυκλώματος.
- ⇒ **Σύνθεση, Χρονική Ανάλυση, Τοποθέτηση και Διασύνδεση:** μεταφορά του κυκλώματος από τη VHDL σε μάσκα κατασκευής για έναν πυρήνα πυριτίου στα 0.18 μm .

Το υποσύστημα ΑΠΚ επιλέχθηκε σαν ένα πολύ καλό παράδειγμα υλοποίησης ενός τμήματος ενός Συστήματος-σε-Chip (System on a Chip). Πρόκειται για ένα κύκλωμα σημαντικής πολυπλοκότητας το οποίο σχεδιάστηκε ακολουθώντας τη μεθοδολογία (Design Flow)

για σύγχρονα ψηφιακά συστήματα, χρησιμοποιώντας τη σημερινή (2001-2002) τεχνολογία αιχμής (0.18 μm).

1.4 Οργάνωση του παρόντος κειμένου

Το κείμενο αυτό ξεκινάει από μια γενική εισαγωγή στο σύστημα MPEG-II (κεφάλαιο 2), ενώ μια πιο συγκεκριμένη όψη του υποσυστήματος Αντιστάθμισης Πληροφορίας Κίνησης παρουσιάζεται στο κεφάλαιο 3.

Η κυρίως υλοποίηση περιγράφεται στα κεφάλαια 4 (κατάτμηση του κυκλώματος και περιγραφή του), 5 (εργαλεία που χρησιμοποιήθηκαν, προβλήματα που αντιμετωπίστηκαν) και 6 (τελικό κύκλωμα και επιδόσεις του).

Στο κεφάλαιο 7 επιχειρείται μια επισκόπηση της εμπειρίας που αποκτήθηκε και των συμπερασμάτων που αποκομίστηκαν. Παραπομπές σε κείμενα για περισσότερες λεπτομέρειες βρίσκονται στο τέλος, στο κεφάλαιο 8.

2

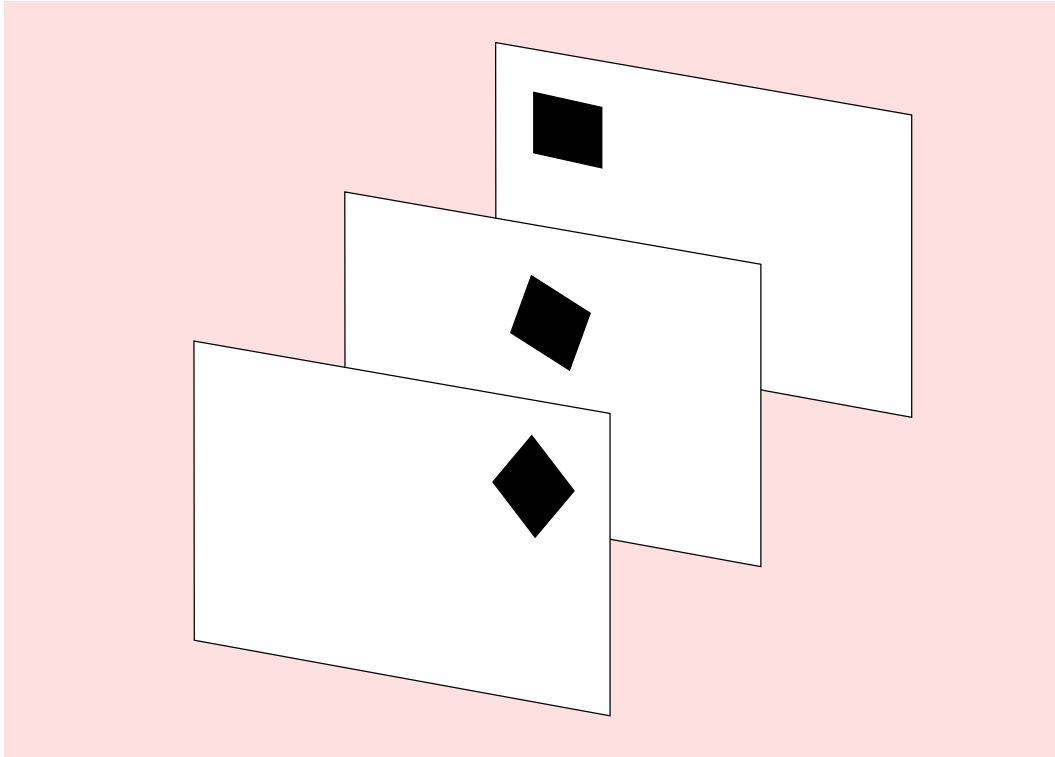
Το σύστημα κωδικοποίησης MPEG-II

2.1 Βασική αρχή

Η ιδέα της κωδικοποίησης της οπτικοακουστικής πληροφορίας βασίζεται στην συστηματική κατανόησή της τελευταίας, με στόχο την παρατήρηση κάποιας προβλέψιμης συμπεριφοράς. Η βασική αρχή για τη μετάδοση μικρότερης ποσότητας δεδομένων είναι ότι αν μεταδοθούν μόνο οι “αλλαγές” μεταξύ διαδοχικών εικόνων αντί για ολόκληρες τις εικόνες, θα έχουμε κερδίσει ένα μεγάλο μέρος του εύρους ζώνης.

Στο σχήμα 2.1 φαίνεται ένα παράδειγμα για ένα μέρος από πληροφορία video που πρέπει να κωδικοποιήσουμε. Η παλαιότερη χρονικά εικόνα βρίσκεται στο πίσω μέρος και η πιο καινούρια στο μπροστινό. Στο video αυτό διακρίνουμε ένα μικρό μαύρο τετράγωνο, το οποίο κινείται προς τα δεξιά και ταυτόχρονα περιστρέφεται προς τα δεξιά. Η μετάδοση χωρίς κωδικοποίηση θα ήταν ισοδύναμη με ολόκληρη τη μετάδοση της πιο πίσω εικόνας σε πρώτη φάση, ολόκληρη τη μετάδοση της μεσαίας και τέλος ολόκληρη τη μετάδοση της μπροστινής.

Στην κωδικοποίηση MPEG-II, η επιτυχής συμπίεση των δεδομένων γίνεται σε δύο ταυτόχρονες κατευθύνσεις: περιορίζοντας το εύρος ζώνης για τη μετάδοση μεμονωμένων κομ-

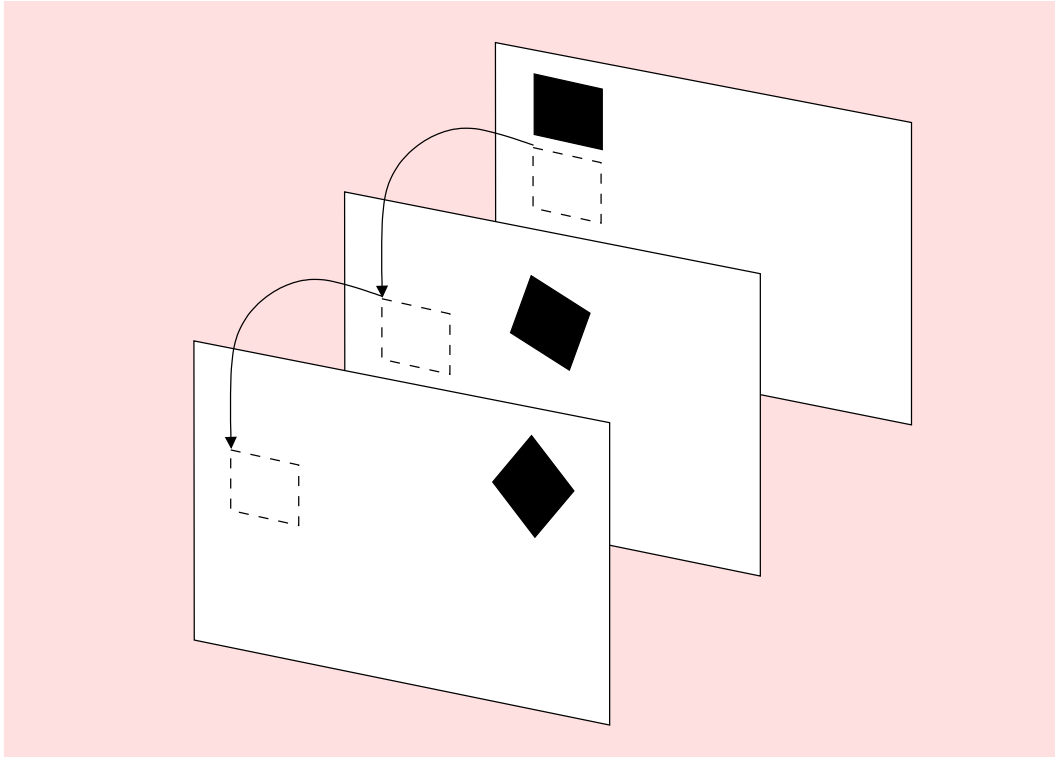


Σχήμα 2.1: Απλή μετάδοση video

ματιών εικόνων και εκτιμώντας την κίνηση μεταξύ διαδοχικών εικόνων.

Όταν χρειάζεται η μεταφορά ενός κομματιού μιας εικόνας, αυτό δε γίνεται με τον παραδοσιακό τρόπο της μεταφοράς των τιμών των pixels της – σαρώνοντας δηλ. την εικόνα και μεταδίδοντας κάθε pixel. Αντίθετα, το κομμάτι εκείνο μεταφέρεται στο διδιάστατο πεδίο των συχνοτήτων. Εκεί ο κωδικοποιητής έχει τη δυνατότητα να αποκόψει τις αρκετά υψηλές συχνότητες, πράγμα που μειώνει την αντίθεση της εικόνας. Ανάλογα με την ποιότητα που πρέπει να επιτευχθεί ο κωδικοποιητής μπορεί να αφήσει τον κατάλληλο αριθμό συχνοτήτων. Επιπρόσθετα, η μεταφορά κομματιών εικόνων ακολουθείται και από αλγόριθμους τύπου Huffman που καταφέρνουν μια επιπλέον συμπίεση.

Η βασικότερη καινοτομία που εισάγει το πρότυπο MPEG-II είναι η προβλεψη της κίνησης ανάμεσα στις εικόνες – αντίθετα, η αναπαράσταση των στάσιμων εικόνων σε μορφή συχνοτήτων ακολουθεί τις κωδικοποιήσεις JPEG. Εδώ, ο κωδικοποιητής προσπαθεί να εντοπίσει τις ομοιότητες που υπάρχουν από εικόνα σε εικόνα εστιάζοντας στην εύρεση αντικειμένων στάσιμων ή και κινούμενων. Η εύρεση των περιοχών αυτών (όπως και όλος ο αλγόριθμος MPEG-II) γίνεται με βάση κομμάτια των 16×16 pixels που ονομάζονται macroblocks. Εάν ένα macroblock από μια εικόνα βρεθεί να είναι παρόμοιο με κάποια άλλη, αντί να μεταδίδεται από την αρχή η εικόνα μεταδίδονται διανύσματα που ορίζουν από ποιά



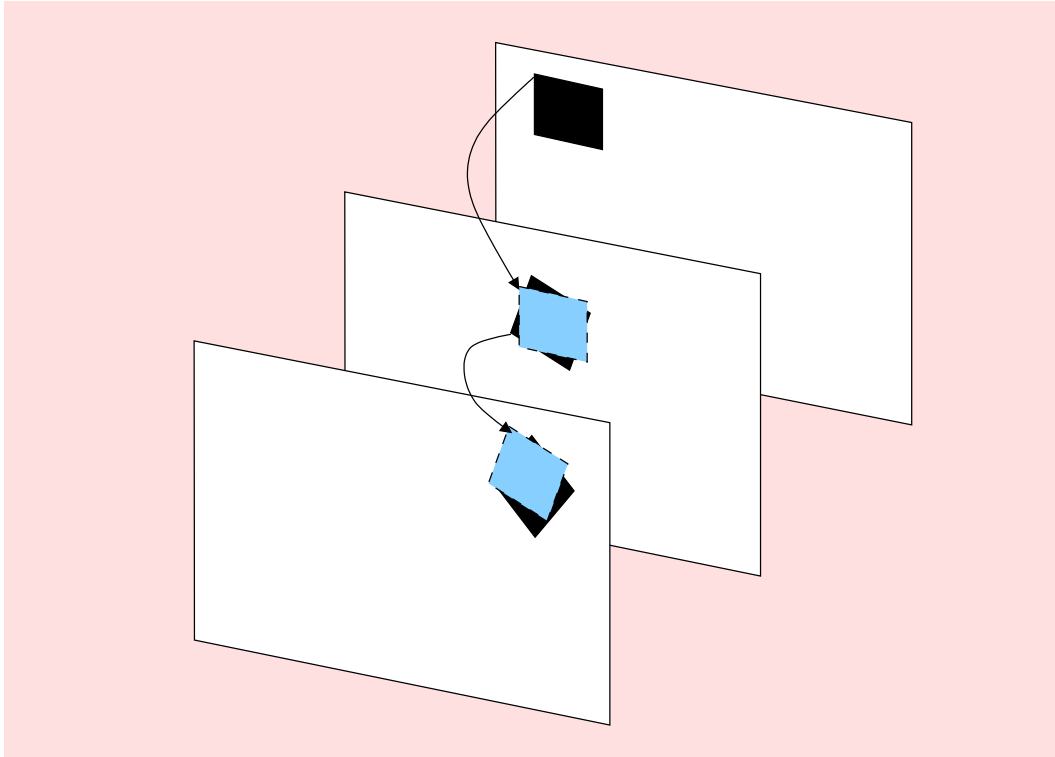
Σχήμα 2.2: Πρόβλεψη στάσιμης περιοχής

εικόνα και από ποιο μέρος αυτής προέρχεται η πληροφορία.

Στο σχήμα 2.2 φαίνεται αυτή η εφαρμογή για το άσπρο μέρος των εικόνων. Όταν υπάρχουν ακίνητα τοπία ή στάσιμες εικόνες στο χρόνο, η εφαρμογή αυτής της μεθόδου είναι άμεση: χρειαζόμαστε μόνο τα διανύσματα κίνησης. Το πρότυπο όμως ορίζει και την εφαρμογή των παραπάνω ακόμα και σε μη τέλει περιπτώσεις. Αν η πληροφορία που θέλουμε να αντιγράψουμε από μια γειτονική εικόνα δεν είναι ακριβώς η ίδια (αλλά μοιάζει αρκετά), την αντιγράφουμε και ταυτόχρονα μεταδίδουμε σαν εικόνα τη διαφορά του αντιγραμμένου κομματιού με το επιθυμητό. Αυτό είναι επίσης κερδοφόρο αφού η διόρθωση θα μεταφέρει πολύ λιγότερη πληροφορία από ολόκληρη την περιοχή.

Σχηματικά, στην εικόνα 2.3 φαίνεται η διαδικασία αυτή για τη σειρά εικόνων που είχαμε προηγουμένως. Για να περιγράψουμε το κινούμενο (αλλά και περιστρεφόμενο) ορθογώνιο ορίζουμε την περιοχή και την εικόνα πηγής (πίσω εικόνα), και παρέχεται η διόρθωση από την αντιγραφή που θα κάνουμε.

Μια πολύ καλή εισαγωγή στο MPEG-II αποτελούν τα [3] και [7]. Για περισσότερες πληροφορίες διάφορα κείμενα βρίσκονται στο δίκτυο. Μια αρχική σελίδα βρίσκεται στο [6]. Ολοκληρωμένες παρουσιάσεις για το MPEG-II μπορούν να βρεθούν και στα δύο βιβλία [4] και [5].



Σχήμα 2.3: Πρόβλεψη κινούμενης περιοχής

2.2 Η δομή ενός ρεύματος MPEG-II

Ένα ρεύμα MPEG-II (MPEG-II stream) περιέχει ένα μεγάλο πλήθος από διαφορετικά στοιχεία. Πρόσθετα σε αυτά που γράφονται παραπάνω για τη μορφή του ρεύματος εικόνων, προβλέπεται ένας αριθμός στοιχείων που καταλήγουν στον τελικό χρήστη. Πληροφορία ήχου, υπορεύματα MPEG-II για περισσότερα του ενός videos, πληροφορία καναλιών και Teletext, προβλεπόμενες θέσεις για το χρήστη είναι ένα δείγμα της δομής ενός καναλιού MPEG-II.

Το υποσύστημα ΑΠΚ που εξετάζουμε, είναι υπεύθυνο για την αποκωδικοποίηση μέρους της πληροφορίας video. Αυτή βρίσκεται στο μέρος του ρεύματος MPEG-II που ονομάζεται Video Elementary Stream (Video ES). Τα στοιχειώδη ρεύματα Video και Audio ES εξάγονται από το συνολικό ρεύμα όπου υπάρχουν με τη μορφή πακέτων PES (Packetized Elementary Stream). Τα πακέτα PES συγκολλούνται από τον αποκωδικοποιητή στη σωστή τους σειρά.

Αυτά τα πακέτα βρίσκονται στις κορυφαίες δομές του ρεύματος, οι οποίες είναι δύο: το Program Stream ή το Transport Stream. Αυτά μεταδίδονται κατά περίπτωση: το Program Stream περιέχει πληροφορία MPEG-II για εφαρμογές που διαθέτουν ευέλικτο εύρος ζώ-

νης, όπως π.χ. ένας δίσκος DVD. Το Transport Stream έχει σταθερό εύρος ζώνης και χρησιμοποιείται για τη μετάδοση από σταθμούς ψηφιακής τηλεόρασης.

Η πλήρης περιγραφή του ρεύματος MPEG-II βρίσκεται στο πρότυπο του MPEG-II για ολόκληρο το σύστημα [2].

2.3 Η δομή ενός Στοιχειώδους Ρεύματος Video

Ένα στοιχειώδες ρεύμα Video αποτελείται από μια συνεχή ροή ομάδων εικόνων (Group of Pictures ή GOP). Η μορφή αυτών των ομάδων εικόνων επιλέγεται από τον κωδικοποιητή με βάση τα κριτήρια που τίθενται προς αυτόν για το συγκεκριμένο video και την ποιότητα και εύρος ζώνης που πρέπει να επιτύχει. Οι ομάδες εικόνων αποτελούνται από τρία είδη εικόνων.

- ⇒ **Εικόνες I:** Είναι εικόνες που χρησιμοποιούνται σαν αρχική αναφορά για το ρεύμα. Περιέχουν την κανονική πληροφορία της εικόνας (τις πραγματικές τιμές των pixels) και τίποτα άλλο. Εξ' αιτίας αυτού, είναι και οι πιο μεγάλες σε μέγεθος και τυπικά χρησιμοποιούνται με συχνότητα 1 φορά ανά 15 εικόνες.
- ⇒ **Εικόνες P:** Οι εικόνες P δεν περιέχουν πάντα αυθεντική πληροφορία της εικόνας. Αν ο κωδικοποιητής ανιχνεύσει κάποιο τμήμα (macroblock των 16 x 16 pixels) που είναι ίδιο από μια παλαιότερη εικόνα I ή P, τότε ορίζει ένα διάνυσμα που το περιγράφει και δεν δίνει εκ νέου τα pixels της. Αν πάλι αυτό δεν είναι εφικτό, ορίζει κανονικά τα pixels όπως και στις εικόνες I. Οι εικόνες P καταλήγουν περίπου στο μισό μέγεθος από τις I. Τυπικά έχουμε 4 στις 15 εικόνες να είναι τύπου P.
- ⇒ **Εικόνες B:** Τέλος, οι περισσότερες εικόνες είναι τύπου B. Αυτές μπορούν να έχουν αυθεντική πληροφορία εικόνας, αλλά μόνον αν δεν είναι εφικτό να βρει ο κωδικοποιητής κάτι καλύτερο. Το καλύτερο είναι ένα τμήμα εικόνας παρόμοιο με κάποια εικόνα I ή P, είτε από παλαιότερη είτε από πιο καινούρια εικόνα. Μπορούν να ορισθούν 1 ή 2 διανύσματα (παλαιότερης εικόνας, πιο καινούριας εικόνας ή και τα δύο) όπως επίσης και διορθωτική πληροφορία με μορφή pixels. Οι εικόνες B έχουν περίπου το 1/4 του μεγέθους των εικόνων I και τις βρίσκουμε με συχνότητα περίπου 10 στις 15.

Μια περίληψη των παραπάνω στοιχείων απεικονίζεται στον πίνακα 2.1.

Σημειώνεται εδώ ότι λόγω της εξαρτήσεως των εικόνων B τόσο από καινούριες όσο και παλιές εικόνες I ή P, ο αποκωδικοποιητής οφείλει να αποκωδικοποιεί τις εικόνες I ή P πολλές φορές εκτός σειράς και να τις εμφανίζει στην οθόνη μετά εντός σειράς.

| Εικόνα | Πληροφορία pixels | Πρόβλεψη | Διανύσματα | Συχνότητα |
|--------|---------------------|------------|------------|------------|
| I | Κανονική | Όχι | – | 1 στις 15 |
| P | Κανονική | Εμπρόσθια | Από I ή P | 4 στις 15 |
| B | Διόρθωση ή κανονική | Αμφίπλευρη | Από I ή P | 10 στις 15 |

Πίνακας 2.1: Είδη εικόνων σε ένα Video ES

Η πλήρης περιγραφή του στοιχειώδους ρεύματος video βρίσκεται στο πρότυπο του MPEG-II για την πληροφορία video [1].

2.4 Macroblocks και κωδικοποίηση 4:2:0

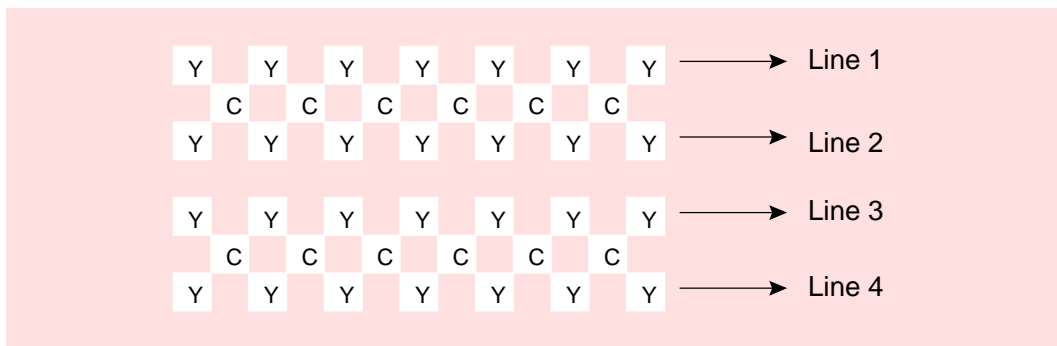
Η κατάτμηση των εικόνων που χρησιμοποιείται από το πρότυπο του MPEG-II ορίζεται στα κομμάτια των 16 x 16 pixels που ονομάζονται macroblocks. Τόσο η πρόβλεψη περιοχών μέσω διανυσμάτων κίνησης όσο και η πληροφορία σε pixels γίνεται τμηματικά στις εικόνες ανά macroblock.

Το σύστημα χρωμάτων που χρησιμοποιεί το MPEG-II είναι το YCrCb αντί του απλοϊκού RGB. Οι δύο αυτοί χώροι προκύπτουν ο ένας από τον άλλο με έναν απλό γραμμικό μετασχηματισμό. Η χρήση του YCrCb επιτρέπει την πιο εύκολη μετάδοση της πληροφορίας διόρθωσης, αφού σε αυτόν τον χώρο τα χρώματα που είναι πιο κοντά στο ανθρώπινο μάτι είναι πιο κοντά και σαν απόσταση μεταξύ των τριάδων των pixels.

Ένα πιο σημαντικό όφελος από την χρήση του YCrCb είναι ότι σε ένα macroblock δεν αποστέλλονται οι τιμές των pixels και για τα τρία συστατικά. Το ανθρώπινο μάτι καταλαβαίνει πολύ καλά τις διακυμάνσεις της φωτεινότητας (που αναπαρίσταται από το συστατικό Y) αλλά είναι λιγότερο ευαίσθητο στις διακυμάνσεις των άλλων δύο. Αυτό που γίνεται στην κωδικοποίηση 4:2:0 είναι ότι αποστέλλεται η τιμή του Y για κάθε pixel, αλλά οι τιμές των Cb και Cr δίνονται για ένα pixel ανά τετράδα (γειτονικά 2 x 2 pixels).

Το παραπάνω φαίνεται και στο σχήμα 2.4, όπου τα Y αντιπροσωπεύουν σημεία της εικόνας που περιέχουν φωτεινότητα, ενώ τα C σημεία όπου περιέχονται χρωματικά συστατικά. Στα σημεία Y η τιμή της φωτεινότητας είναι αυτή του συγκεκριμένου pixel. Στα C η τιμή των δύο χρωμάτων είναι ο μέσος όρος των 4 γειτονικών pixels.

Μια πλήρης περιγραφή της κωδικοποίησης 4:2:0 βρίσκεται στο κεφάλαιο 6.1.1.8 (“4:2:0 Format”) του [1].



Σχήμα 2.4: Σύστημα χρωμάτων 4:2:0

3

Το υποσύστημα ΑΠΚ

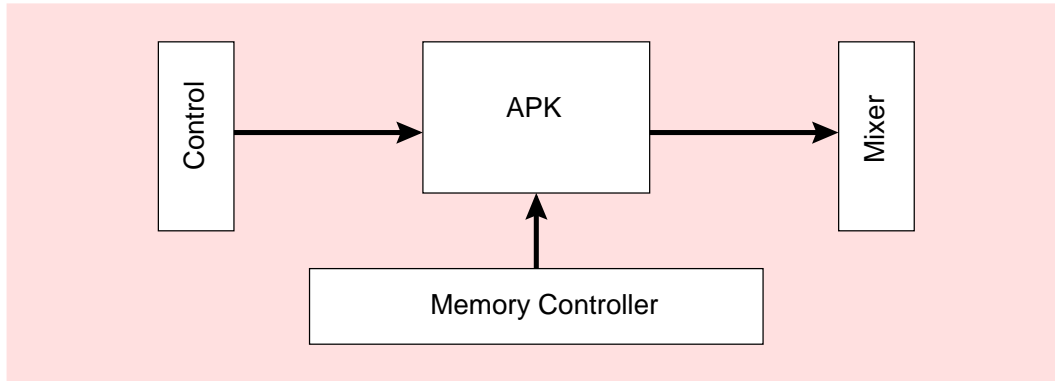
3.1 Περιβάλλον του υποσυστήματος

Το υποσύστημα Αντιστάθμισης Πληροφορίας Κίνησης (ΑΠΚ) βρίσκεται στην καρδιά του αποκωδικοποιητή MPEG-II για στοιχειώδη ρεύματα video. Αποτελεί το ένα από τα δύο βασικά του υποσυστήματα: το ένα αποκωδικοποιεί την πληροφορία των pixels (μέσω αντίστροφου μετασχηματισμού Fourier), ενώ το ΑΠΚ αποκωδικοποιεί την πληροφορία των διανυσμάτων της κίνησης.

Όπως φαίνεται στο σχήμα 3.1, το υποσύστημα επικοινωνεί με 3 άλλα γειτονικά του: το μπλοκ ελέγχου, τον ελεγκτή της μνήμης και τον μίκτη της τελικής εικόνας.

Η επικοινωνία με το μπλοκ ελέγχου γίνεται κυρίως για την προμήθευση των διανυσμάτων κίνησης που θα αποκωδικοποιηθούν. Το μπλοκ ελέγχου μπορεί να αποστέλλει διανύσματα και για μελλοντικά macroblocks, πάντα με τη σωστή σειρά. Το ΑΠΚ τα αποθηκεύει μια εσωτερική του ουρά και τα διαχειρίζεται όταν μπορεί. Άλλες πληροφορίες που προκύπτουν από το μπλοκ ελέγχου είναι γενικότερης φύσεως, όπως ο τύπος του συγκεκριμένου macroblock, η πρόβλεψη που πρέπει να γίνει κλπ.

Ο ελεγκτής της μνήμης είναι αυτός που θα δώσει στο ΑΠΚ τα παράθυρα (ευρύτερες πε-



Σχήμα 3.1: Περιβάλλον του ΑΠΚ

ριοχές της εικόνας) που ορίζονται από τα διανύσματα κίνησης. Απο αυτά τα ευρύτερα παράθυρα το ΑΠΚ θα επιλέξει τις ακριβείς περιοχές που θα χρειαστεί για την επεξεργασία των pixels που προκύπτουν από την πρόβλεψη της κίνησης.

Τέλος, ο μίκτης είναι αυτός που θα παραλάβει την αντισταθμισμένη εικόνα από την πρόβλεψη που έχει κάνει το ΑΠΚ, και θα τη συνδυάσει με την πληροφορία που έχει προκύψει από τον αποκωδικοποιητή της πληροφορίας pixels ώστε να παραχθεί η τελική εικόνα.

Το ΑΠΚ λειτουργεί με εικόνες της μορφής 4:2:0. Στην επεξεργασία, αυτό επιτυγχάνεται με το χειρισμό πρώτα της πληροφορίας της φωτεινότητας (Y) για όλο το macroblock (16 x 16 pixels), και κατόπιν για ένα-ένα τα συστατικά της φωτεινότητας (Cb, Cr) που όμως σε μέγεθος είναι 8 x 8 pixels το καθένα για όλο το macroblock.

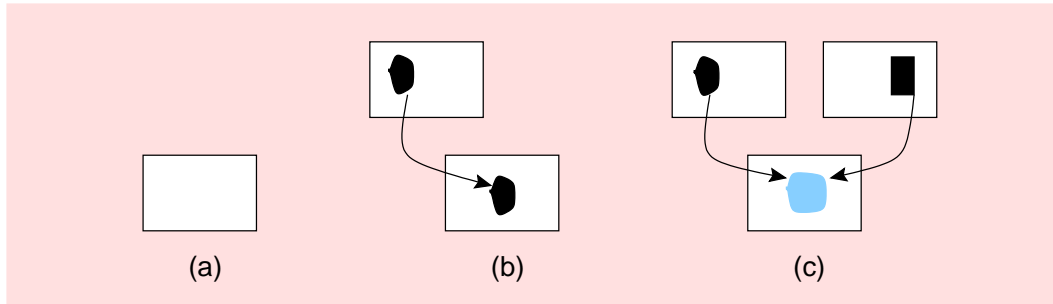
3.2 Τύποι εικόνων και προβλέψεων

Η αντιστάθμιση της πληροφορίας της κίνησης λειτουργεί με ένα πλήθος από παράμετρους. Στις επόμενες υποενότητες θα γίνει μια κατηγοριοποιημένη παρουσίασή τους.

3.2.1 Αριθμός εικόνων πηγής

Τα macroblocks που πρέπει να αποκωδικοποιηθούν έχουν έναν αριθμό εικόνων πηγής. Μια κατηγοριοποίηση με τέτοια βάση ορίζει τους παρακάτω τρόπους λειτουργίας του ΑΠΚ.

- ⇒ **Εικόνες χωρίς διανύσματα κίνησης:** το ΑΠΚ στην ουσία δεν θα πρέπει να λειτουργήσει. Αυτός ο τρόπος είναι χρήσιμος π.χ. για εικόνες τύπου I, όπου δεν υπάρχει πρόβλεψη κίνησης ή για κάποια macroblocks από εικόνες P ή B που δεν προκύπτουν



Σχήμα 3.2: Αριθμός εικόνων πηγής

από πληροφορία κίνησης. Το ΑΠΚ πρέπει να διατηρήσει έναν ορθό τρόπο χειρισμού τέτοιων macroblocks και σαν αποτέλεσμα πρέπει να εξάγει μια εικόνα με μηδενικές τιμές pixels. Ο στόχος του τελευταίου είναι ο μίκτης να χειρίζεται τις εικόνες χωρίς να ξέρει περαιτέρω λεπτομέρειες – απλά προσθέτοντας την εικόνα που προκύπτει από το ΑΠΚ και αυτήν από τον αποκωδικοποιητή των pixels.

- ⇒ **Εικόνες με εμπρόσθια ή οπίσθια πρόβλεψη κίνησης:** ορίζεται μία εικόνα πηγής. Το ΑΠΚ επιλέγει και επεξεργάζεται τα διανύσματα που ορίζουν τη συγκεκριμένη υποπεριοχή και εξάγει την εικόνα που προκύπτει.
- ⇒ **Εικόνες με αμφίπλευρη πρόβλεψη κίνησης:** ορίζονται δύο εικόνες πηγής. Το ΑΠΚ πρέπει να επεξεργαστεί ανεξάρτητα και παράλληλα αυτές τις δύο διαφορετικές εικόνες. Σαν αποτέλεσμα, η εικόνα που προκύπτει είναι ο μέσος όρος των 2 εικόνων pixel προς pixel.

Στο σχημα 3.2 φαίνονται οι 3 τρόποι λειτουργίας με βάση αυτήν την παράμετρο. Στο (a) δεν υπάρχει εικόνα πηγής, στο (b) υπάρχει μία εικόνα πηγής και στο (c) δύο.

3.2.2 Τρόποι πρόβλεψης

Οι εικόνες που χρησιμοποιούνται στα σήματα τηλεόρασης και video μπορεί να ανήκουν σε δύο κατηγορίες: εικόνες Frame και εικόνες Field. Η κατηγοριοποίηση αυτή βασίζεται στην παραδοσιακή σάρωση των τηλεοράσεων. Αν η σάρωση γίνεται γραμμή προς γραμμή προκύπτουν εικόνες Frame. Αν η σάρωση γίνεται πρώτα για τις περιττές γραμμές της εικόνας και ύστερα για τις άρτιες, προκύπτουν εικόνες Field.

Στο MPEG-II τα πράγματα περιπλέκονται περαιτέρω μέσω του τρόπου που γίνεται η πρόβλεψη στον κωδικοποιητή. Έτσι, Ανάλογα και με τον τύπο της εικόνας και της πρόβλεψης, ορίζονται 4 κατηγορίες λειτουργίας του ΑΠΚ.

- ⇒ **Εικόνα Frame με πρόβλεψη Frame:** ο κωδικοποιητής έχει χρησιμοποιήσει αρχική εικόνα τύπου Frame για την πρόβλεψη, ενώ η τελική εικόνα θα είναι ομοίως τύπου Frame. Έτσι, ένα διάνυσμα αρκεί για να περιγράψει το macroblock που θα χρησιμοποιηθεί για την πρόβλεψη.
- ⇒ **Εικόνα Frame με πρόβλεψη Field:** είμαστε σε μια πιο πολύπλοκη κατάσταση. Η τελική μας εικόνα είναι τύπου Frame, αλλά το αρχικό σήμα του κωδικοποιητή ήταν τύπου Field. Επομένως, ορίζονται 2 διανύσματα για το τελικό macroblock που περιγράφουν τις περιττές και τις άρτιες γραμμές πηγής ξεχωριστά.
- ⇒ **Εικόνα Field με πρόβλεψη Frame:** η τελική μας εικόνα είναι τύπου Field, δηλαδή είμαστε σε φάση που ασχολούμαστε με macroblocks αποκλειστικά άρτιων ή αποκλειστικά περιττών γραμμών. Ο κωδικοποιητής όμως είχε δουλέψει σε εικόνες τύπου Frame για να κάνει την πρόβλεψη και επομένως η πρόβλεψη αφορά μόνο κομμάτια 8 γραμμών (οι άρτιες ή οι περιττές του αρχικού). Έτσι, στο ΑΠΚ δίνονται και πάλι δύο διανύσματα, που αυτή τη φορά αφορούν το πάνω και το κάτω μέρος του macroblock (τις πάνω 8 και τις κάτω 8 γραμμές).
- ⇒ **Εικόνα Field με πρόβλεψη Field:** η εικόνα Field που θα παραχθεί έχει προβλεφθεί με βάση μια αρχική εικόνα Field στον κωδικοποιητή. Έτσι, υπάρχει 1 διάνυσμα που μας περιγράφει ολόκληρη την περιοχή πηγής.

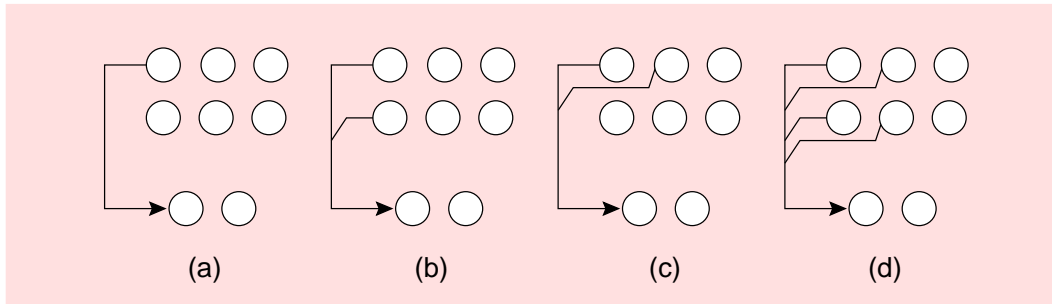
Πρέπει να σημειωθεί ότι το πρότυπο MPEG-II ορίζει και τον τρόπο πρόβλεψης Dual Prime, που όμως δεν χειρίζεται το ΑΠΚ.

3.2.3 Φίλτρο μισού pixel

Τα διανύσματα που περιγράφουν τις εικόνες πηγής διαθέτουν και ένα δυαδικό ψηφίο σταθερής υποδιαστολής. Αυτό έχει χρησιμοποιηθεί από το πρότυπο MPEG-II για την αποτελεσματικότερη περιγραφή σκηνών κινούμενων αντικειμένων. Σαν σύλληψη ορίζει ότι η περιοχή που αντιγράφουμε από την προηγούμενη ή επόμενη εικόνα θα απέχει και μισό pixel παραπάνω.

Στην πράξη, η μετακίνηση μισού pixel σημαίνει ότι η περιοχή που προκύπτει είναι το ημίθροισμα pixel προς pixel των γειτονικών σημείων που προκύπτουν. Εάν για παράδειγμα έχουμε ένα διάνυσμα που απέχει 2.5 pixels, η αντιγραφή για το πρώτο μας pixel θα είναι το ημίθροισμα των pixels 2 και 3, το δεύτερο το ημίθροισμα των pixels 3 και 4 κ.ο.κ.

Το σχήμα 3.3 δείχνει μια μεταφορά χωρίς φίλτρο μισού pixel (a), μια με κάθετο φίλτρο (b), μια με οριζόντιο (c) και μια και με τα δύο (d).



Σχήμα 3.3: Φίλτρο μισού pixel

Το υποσύστημα ΑΠΚ πρέπει να χειρίζεται τέτοιου είδους φίλτρα, που ορίζονται και για τις δύο διαστάσεις (κάθετη και οριζόντια) σε ανεξάρτητη βάση. Ανάλογη δέσμευση στοιχείων μνήμης πρέπει να υπάρχει ώστε να υποστηρίζεται αποτελεσματικά τέτοιο φιλτράρισμα στις περιοχές πηγής (χωρίς επιπλέον χρόνο από τον κανονικό).

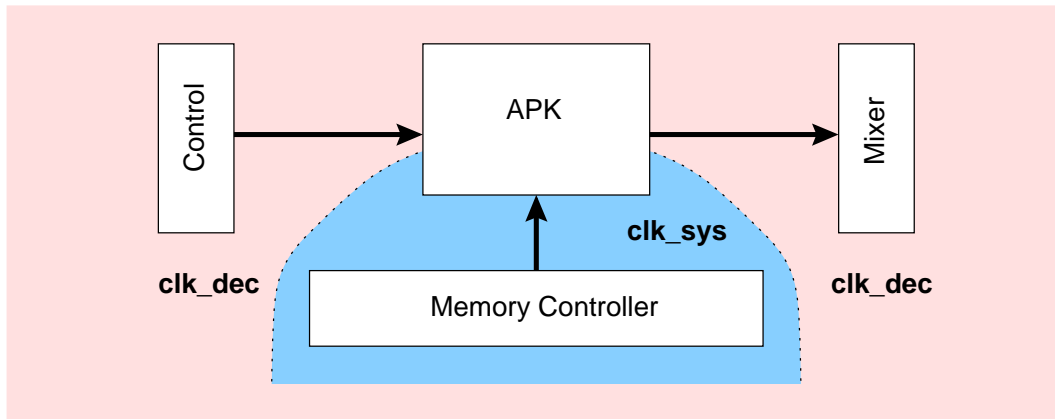
3.3 Επεξεργασία διανυσμάτων

Τα διανύσματα που έρχονται στο ΑΠΚ υπόκεινται σε μια επεξεργασία. Ο κεντρικός έλεγχος στέλνει έναν αριθμό διανυσμάτων, που αναφέρεται στα διανύσματα της φωτεινότητας του συγκεκριμένου macroblock, και έρχονται κατ' ευθείαν από το ρεύμα MPEG-II.

Το ΑΠΚ διαχωρίζει τα μέρη των διανυσμάτων που αφορούν την ευρύτερη περιοχή και βάση αυτών ζητάει τα σωστά παράθυρα από τη μνήμη. Επιπλέον, εκτελεί σε αυτά μια σειρά πράξεων ανάλογα με το είδος της εικόνας και βάση αυτών των πράξεων δημιουργούνται τα (μικρότερα σε μέγεθος) διανύσματα που θα οδηγήσουν την πρόβλεψη μέσα στα ευρύτερα παράθυρα που έρχονται από τη μνήμη.

Τέλος, σε αυτήν την επεξεργασία προστίθεται και η ανάγκη παραγωγής των διανυσμάτων για τα χρωματικά συστατικά της εικόνας που θα χρησιμοποιηθούν μετά την περάτωση της επεξεργασίας της φωτεινότητας.

Οι πράξεις για την επεξεργασία των διανυσμάτων είναι κατά το κύριο μέρος τους εξαγωγή πεδίων από τα αρχικά διανύσματα, που ελέγχεται με βάση την εικόνα και τον τρόπο της πρόβλεψης. Οι πράξεις ορίζονται πλήρως στο κεφάλαιο 7.6.3 (“Motion Vectors”) του [1].



Σχήμα 3.4: Ρολόγια του ΑΠΚ

3.4 Μορφή εξόδου προς το μίκτη

Η έξοδος της τελικής εικόνας προς το υποσύστημα του μίκτη οφείλει να υπακούει σε συγκεκριμένους περιορισμούς. Πρέπει να είναι οργανωμένη κατά στήλες και η ροή να είναι ελεγχόμενη από αυτόν.

Η οργάνωση της τελικής εικόνας κατά στήλες είναι μια απαίτηση για την απλοποίηση του μίκτη. Επειδή το υποσύστημα πληροφορίας rixels χρησιμοποιεί αντίστροφο μετασχηματισμό Fourier που εξαγάγει την εικόνα κατά στήλες, αποφασίστηκε η τελική μετατροπή από τους δύο τρόπους της μετάδοσης να γίνεται στο ΑΠΚ.

Για ανάλογους λόγους, η ροή τελικά (μέσω του μίκτη) ελέγχεται από το πόσο γρήγορα έρχονται τα rixels από το υποσύστημα πληροφορίας rixels. Έτσι, άπαξ και το ΑΠΚ δηλώσει ότι έχει ένα macroblock προς μετάδοση η ροή ελέγχεται πλέον από το μίκτη. Το ΑΠΚ δεν έχει δικαίωμα για το συγκεκριμένο macroblock να την σταματήσει έστω και για έναν κύκλο ρολογιού.

3.5 Ρολόγια

Το σύστημα του αποκωδικοποιητή λειτουργεί με δύο διαφορετικά ρολόγια. Αυτό προκύπτει από την ανάγκη του να είναι προσιτός σε διαφορετικά περιβάλλοντα. Έτσι, τα μέρη που σχετίζονται με είσοδο/έξοδο προς το σύστημα πρέπει να λειτουργούν σε ένα ευρύ φάσμα από διαφορετικά ρολόγια, ενώ τα μέρη που κάνουν την κυρίως δουλειά έχουν το δικό τους σταθερό ρολόι.

Στο μπλοκ ΑΠΚ αυτό αντικατοπτρίζεται στη σχέση του με τον ελεγκτή μνήμης. Το ρολόι

αυτής της επικοινωνίας είναι ανεξάρτητο από το ρολόι επικοινωνίας του πυρήνα του, που χρησιμοποιείται επίσης και για την επικοινωνία με τον μίκτη και τον κεντρικό έλεγχο. Στο σχήμα 3.4 φαίνονται τα δύο πεδία ρολογιού, όπου με `clk_mem` αναπαρίσταται το ρολόι της επικοινωνίας με τη μνήμη και με `clk_dec` το κυρίως ρολόι του αποκωδικοποιητή.

4

Μελέτη και σχεδίαση

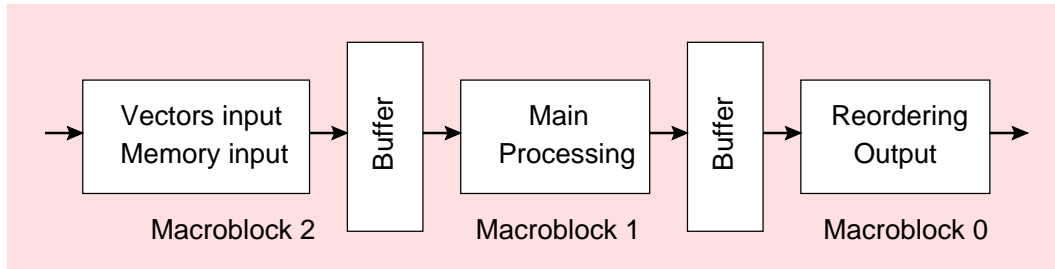
4.1 Ανάλυση απαιτήσεων

Το υποσύστημα ΑΠΚ οφείλει να ικανοποιεί ένα πλήθος απαιτήσεων, τόσο από πραγματικής χρονικής πλευράς (διαφορετικά ρολόγια) όσο και χρονισμών σε κύκλους ρολογιού για την σωστή αποκωδικοποίηση συνεχούς ροής εικόνων.

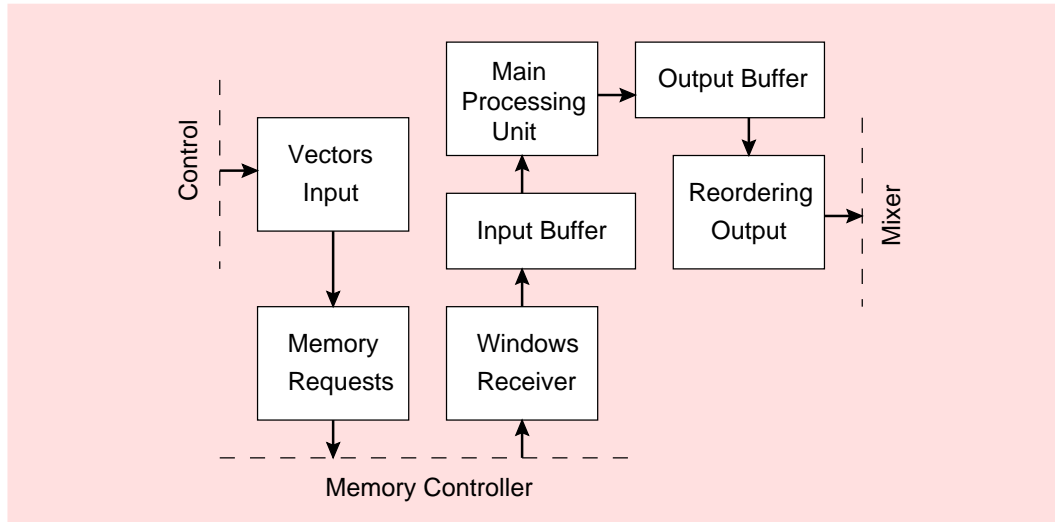
Ο πρώτος περιορισμός ωθεί στη χρήση δύο καλά διαχωρισμένων περιοχών για τα διαφορετικά ρολόγια του κυκλώματος. Έτσι, στον καταμερισμό των μπλοκ ένας σημαντικός παράγοντας είναι το πεδίο συχνότητας που λειτουργούν οι καταχωρητές. Κατά φυσικό επακόλουθο, τα μπλοκ που σχετίζονται με τον ελεγκτή της μνήμης βρίσκονται σε διαφορετικό ρολόι από τα υπόλοιπα.

Ο δεύτερος περιορισμός είναι κάπως πιο πολύπλοκος. Αυτό που απαιτείται από το κύκλωμα τελικά οδηγεί στην απαίτηση να επεξεργαζόμαστε το κάθε macroblock σε πάρα πολύ λίγους κύκλους ρολογιού. Αντί αυτού, υλοποιείται τελικά μια πολύ πιο αποδοτική μέθοδος που κάνει το ΑΠΚ να συμπεριφέρεται σαν μια μεταβλητή pipeline.

Συγκεκριμένα, χρησιμοποιώντας δύο ενδιάμεσους ενταμιευτές (μνήμες μιας πόρτας εγγραφής και μιας ανάγνωσης), η ροή ενός macroblock μέσα στο υποσύστημα χωρίζεται σε



Σχήμα 4.1: Χρονική ροή των macroblocks



Σχήμα 4.2: Μπλοκς του ΑΠΚ

τρεις φάσεις: της εισδοχής-επεξεργασίας διανυσμάτων και των αιτήσεων στη μνήμη, της πραγματικής επεξεργασίας και της τελικής επεξεργασίας και εξόδου προς τον μίκτη. Σχηματικά αυτό φαίνεται στο σχήμα 4.1 όπου το πρώτο macroblock παραδίδεται στο μίκτη, ενώ την ίδια στιγμή το δεύτερο macroblock επεξεργάζεται και γίνονται οι αιτήσεις προς τη μνήμη για το τρίτο.

4.2 Καταμερισμός σε μπλοκς

Ο καταμερισμός του ΑΠΚ σε επιμέρους κομμάτια φαίνεται στο σχήμα 4.2. Διακρίνονται τα 5 κυρίως μέρη και οι δύο ενδιάμεσοι ενταμιευτές, τα 3 εξωτερικά υποσυστήματα με τα οποία υπάρχει ανταλλαγή πληροφορίας καθώς επίσης και η πορεία επεξεργασίας ενός macroblock.

Πιο συγκεκριμένα, τα επιμέρους κομμάτια και οι αρμοδιότητές τους αναλύονται στις επό-

μενες υποενότητες.

4.2.1 Μπλοκ Εισόδου Διανυσμάτων

Στο συγκεκριμένο κομμάτι του κυκλώματος ξεκινάει η πορεία ενός macroblock. Οι αρμοδιότητες του περιλαμβάνουν την είσοδο των διανυσμάτων, την επεξεργασία τους, την φύλαξη τους σε μια ουρά και την αποστολή τους στα επόμενα μπλοκς του ΑΠΚ.

Τα διανύσματα έρχονται από το μπλοκ ελέγχου με τυχαία σειρά μέσα σε ένα macroblock. Ο έλεγχος προσδιορίζει με συγκεκριμένες διευθύνσεις για ποιο διάνυσμα πρόκειται κάθε φορά. Με μια από αυτές τις διευθύνσεις ορίζεται και η μετάβαση στα διανύσματα του επόμενου macroblock. Τα διανύσματα γράφονται στην κεφαλή της ουράς.

Η επεξεργασία των διανυσμάτων που ορίζει το πρότυπο MPEG-II περατώνεται μέσα σε ένα κύκλο ρολογιού. Κατόπιν, είναι έτοιμα προς χρήση από το υπόλοιπο ΑΠΚ το οποίο τα παραλαμβάνει από το τέλος της ουράς. Μέσω συνεννόησης με τα υπόλοιπα μπλοκς, ο δείκτης στο τέλος της ουράς προχωράει όταν αυτά τελειώσουν το διάβασμα και εφόσον υπάρχουν καινούρια (επεξεργασμένα) διανύσματα.

4.2.2 Μπλοκ Αιτήσεων Μνήμης

Εδώ γίνεται η αναγνώριση του τύπου της εικόνας από τα διανύσματα και οι κατάλληλες αιτήσεις προς τη μνήμη για την αποστολή των ευρύτερων παραθύρων της πρόβλεψης.

Ανάλογα με τον τύπο της πρόβλεψης και της εικόνας, ορίζονται λιγότερα ή περισσότερα διανύσματα, που αντιστοιχούν και σε λιγότερες ή περισσότερες εικόνες. Έτσι, μετά την αναγνώριση του σωστού αριθμού παραθύρων, γίνονται οι αιτήσεις διαδοχικά για τα παράθυρα της φωτεινότητας και κατόπιν για τα συστατικά των χρωμάτων.

Τα παράθυρα που ζητούνται ορίζονται από τα κατάλληλα κάθε φορά μέρη των περισσότερων σημαντικών ψηφίων των διανυσμάτων. Ο τρόπος ζήτησης από τη μνήμη είναι ένα κλασικό πρωτόκολλο request-acknowledge. Ο ελεγκτής της μνήμης μπορεί να δεχθεί πολλές αιτήσεις πριν αρχίσει να απαντάει. Αυτό το χαρακτηριστικό το εκμεταλλευόμαστε κάνοντας τις αιτήσεις όσο το δυνατόν πιο γρήγορα.

4.2.3 Μπλοκ Εισόδου Παραθύρων

Σε αυτό το δεύτερο μπλοκ επικοινωνίας με τον ελεγκτή της μνήμης χειριζόμαστε τις απαντήσεις στις προηγούμενες αιτήσεις. Ο ελεγκτής της μνήμης δεσμεύεται να απαντήσει με

την ίδια σειρά που έγιναν οι αιτήσεις του.

Το συγκεκριμένο μπλοκ υλοποιεί έναν παρόμοιο μηχανισμό αναγνώρισης του πλήθους και του είδους των παραθύρων για το macroblock, ώστε να ξεχωρίζει την πιθανώς εκρηκτική σε κίνηση απάντηση του ελεγκτή μνήμης. Ο τελευταίος έχει τη δυνατότητα να απαντήσει μονοκόμματα σε όλες τις αιτήσεις για ένα macroblock, οπότε πρέπει να είμαστε σε θέση να ξεχωρίζουμε τον σωστό κάθε φορά αριθμό διανυσμάτων και το σωστό πλήθος λέξεων μνήμης που θα έρθουν. Σημειώνεται εδώ ότι οι λέξεις για τη φωτεινότητα είναι περίπου οι διπλάσιες από αυτές για τα χρώματα.

Οι απαντήσεις της μνήμης τοποθετούνται στον πρώτο ενταμιευτή, που ονομάζεται Ενταμιευτής Εισόδου. Η μνήμη που τον υλοποιεί περιέχεται εντός του ολοκληρωμένου κυκλώματος (on-chip memory) και είναι τύπου SRAM με μια πόρτα ανάγνωσης και μια εγγραφής. Αυτό το μπλοκ χρησιμοποιεί μόνο την πόρτα εγγραφής.

4.2.4 Μπλοκ Μονάδας Κυρίως Επεξεργασίας

Το μεγαλύτερο μπλοκ του ΑΠΚ είναι η Μονάδα Κυρίως Επεξεργασίας. Εδώ πρέπει να γίνει όλη η πραγματική επεξεργασία των παραθύρων, ώστε να παραχθούν τα pixels του τελικού macroblock πρόβλεψης.

Το μπλοκ αυτό υλοποιείται από δύο παράλληλους δρόμους δεδομένων: έναν για εμπρόσθια πρόβλεψη και έναν για οπίσθια. Ανάλογα με τον τύπο της πρόβλεψης κανένας από τους δύο, ένας από τους δύο ή και οι δύο θα είναι ενεργοί.

Ο κάθε δρόμος είναι ανεξάρτητος ως προς την επεξεργασία του. Μπορεί να επιλέξει οποιοδήποτε από τα διανύσματα του αντιστοιχεί για τη συγκεκριμένη φάση. Σημειώνεται ότι ανάλογα με την περίπτωση, ένα διάνυσμα μπορεί να χρησιμοποιηθεί και από τους δύο δρόμους αλλά για διαφορετικά παράθυρα μνήμης.

Οι δύο δρόμοι έχουν κοινή αφετηρία από τον Ενταμιευτή εισόδου, ο οποίος παρέχει τα παράθυρα για την πρόβλεψη. Ο Ενταμιευτής εισόδου έχει μια πόρτα ανάγνωσης αφιερωμένη στη Μονάδα Κυρίως Επεξεργασίας, αλλά δημιουργείται το θέμα ότι κάθε δρόμος χρειάζεται διαφορετικά παράθυρα. Για την αποφυγή δημιουργίας μιας καινούριας πόρτας στον Ενταμιευτή, το μπλοκ διαβάζει εναλλάξ από την πόρτα τα δύο παράθυρα, χρησιμοποιώντας το διπλάσιο πλάτος pixels κάθε φορά. Αυτό κάνει λίγο πιο πολύπλοκο τον έλεγχο του κάθε δρόμου, αλλά μειώνει σημαντικά το μέγεθος του συνολικού κυκλώματος.

Οι φάσεις που περνά η Μονάδα Κυρίως Επεξεργασίας είναι 4: το πρώτο μισό της φωτεινότητας του macroblock, το δεύτερο μισό, το χρώμα Cr και το χρώμα Cb. Η φωτεινότητα διασπάται σε δύο μισά ώστε η υλοποίηση του Ενταμιευτή εξόδου να είναι αποτελεσματική.

Για να επιτευχθεί ένας ικανοποιητικός ρυθμός επεξεργασίας χωρίς τεράστια αποθέματα μνήμης, οι δύο ενταμιευτές έχουν δυνατότητα αποθήκευσης μόνον για ένα macroblock ο καθένας. Μόλις υπάρχει χώρος στον καθένα από αυτούς για το επόμενο macroblock, τότε και μόνον τότε επιτρέπεται η εγγραφή του. Στον Ενταμιευτή εισόδου αυτό ελέγχεται από το μπλοκ Αιτήσεων Μνήμης: εάν δεν υπάρχει ελεύθερος χώρος για το συγκεκριμένο παράθυρο, οι αιτήσεις σταματούν και περιμένουν μέχρι να ελευθερωθεί. Στον Ενταμιευτή εξόδου, για να επιτύχουμε μια ανάλογη διακριτοποίηση της πληροφορίας ολόκληρου του macroblock (όπως είχαμε τα φυσικά ξεχωριστά παράθυρα), το χωρίζουμε στις προαναφερθείσες 4 φάσεις. Με αυτόν τον τρόπο και με την κατάλληλη συνεννόηση με το μπλοκ Αναπροσαρμοστικής Εξόδου πετυχαίνεται το ζητούμενο.

Ένα σημαντικό θέμα στη λειτουργία της Μονάδας Κυρίως Επεξεργασίας είναι ο συγχρονισμός των δύο δρόμων. Στην περίπτωση που και οι δύο είναι ενεργοί, η τελική εικόνα προκύπτει από την συμψήφιση των εικόνων των δύο δρόμων rixel προς rixel. Πολλά προβλήματα όμως δημιουργούνται από το γεγονός ότι οι δύο δρόμοι μπορούν να ακολουθούν σημαντικά διαφορετικές τακτικές πρόβλεψης. Π.χ. η εμπρόσθια πρόβλεψη μπορεί να χρησιμοποιεί φίλτρα μισού rixel σε κάθετο επίπεδο, ενώ η οπίσθια όχι. Αυτό οδηγεί την εμπρόσθια πρόβλεψη σε καθυστέρηση μιας γραμμής (ώστε να υπάρχει ο απαιτούμενος χρόνος αποθήκευσης της βοηθητικής γραμμής για το φίλτρο). Τα πολλά προβλήματα τέτοιου είδους συγχρονισμού λύνονται με την αναγωγή του ελέγχου οποιασδήποτε πρόβλεψης στο χρόνο αυτής της χειρότερης περίπτωσης ανά γραμμή.

Με αυτήν την αναγωγή πετυχαίνεται διπλός σκοπός: ο έλεγχος της Μονάδας γίνεται τελικά απλούστερος (αφού δεν υπάρχει εξάρτηση των ελέγχων των δύο δρόμων) και οι περιπτώσεις της Επαλήθευσης του κυκλώματος σε τελικό επίπεδο απλοποιούνται σημαντικά. Και οι δύο αυτοί παράμετροι βοηθούν σε μεγάλο βαθμό το χρόνο ανάπτυξης του κυκλώματος ΑΠΚ.

Η πραγματική επεξεργασία του κάθε δρόμου αποτελείται από ένα αρκετά πολύπλοκο κύκλωμα το οποίο αναλαμβάνει την μεταφορά των rixels του παραθύρου στην σωστή κάθε φορά θέση της τελικής εικόνας, την εφαρμογή των δύο φίλτρων μισού rixel (κάθετου και οριζόντιου) και το σωστό υπολογισμό της περιοχής εξόδου ανάλογα με τον τύπο της τελικής εικόνας. Απο το επίπεδο του σχεδιασμού υπολογίστηκαν οι καθυστερήσεις του συγκεκριμένου κυκλώματος ώστε να χωράει σε έναν κύκλο ρολογιού, πράγμα που επαληθεύτηκε στην τελική φυσική τοποθέτηση του κυκλώματος.

Στην ειδική περίπτωση που το macroblock που επεξεργάζεται δεν έχει πρόβλεψη, ο ρόλος του μπλοκ είναι να παράγει το σωστό αριθμό μηδενικών rixels στην έξοδο. Για να γίνει αυτό, όλες οι FSMs ελέγχου του μπλοκ χρειάζεται να διαθέτουν έναν ειδικό δρόμο που να το υλοποιεί. Αυτοί οι δρόμοι έχουν δημιουργηθεί ώστε να χειρίζονται τη συγκεκριμένη περίπτωση με τρόπο που τα υπόλοιπα μπλοκ να μην καταλαβαίνουν. Η συγκεκρι-

μένη σχεδιαστική απόφαση κινήθηκε για παρόμοιους λόγους με παραπάνω – λιγότερες περιπτώσεις τελικής Επαλήθευσης για όλο το κύκλωμα, αφού αρκεί η συγκεκριμένη περίπτωση να περνάει σωστά μόνον από την Μονάδα Κυρίως Επεξεργασίας.

4.2.5 Μπλοκ Αναπροσαρμοστικής Εξόδου

Στο τελευταίο αυτό μπλοκ του υποσυστήματος ΑΠΚ, πραγματοποιείται η αποστολή της τελικής εικόνας πρόβλεψης στο μίκτη μετά από κατάλληλη αναπροσαρμογή της.

Οι περιπτώσεις που πρέπει να χειριστεί το μπλοκ οριοθετούνται από τον τύπο της εικόνας και τον τρόπο που το υποσύστημα πρόβλεψης rixels έχει ξεκινήσει το μετασχηματισμό του. Πάντα η τελική εικόνα παραδίδεται κατά στήλες (ενώ στον Ενταμιευτή εξόδου βρίσκεται ανά γραμμές), αλλά οι διάφορες περιπτώσεις σχετίζονται με την αποστολή των κατάλληλων γραμμών της εικόνας στη σωστή σειρά.

Το πρωτόκολλο που ακολουθείται για την επικοινωνία με το μίκτη υποχρεώνει το μπλοκ αυτό να υπολογίζει σωστά την αρχική άδεια που του δίνει για να σηματοδοτεί την καινούρια διαθέσιμη εικόνα. Αυτό πρέπει να γίνεται όσο το δυνατόν πιο γρήγορα μεν, αλλά να εξασφαλίζει την απρόσκοπτη αποστολή rixels από εκεί και πέρα. Έτσι, χρειάζεται να υπολογιστεί για κάθε περίπτωση η χρονική στιγμή που η εικόνα δεν είναι πλήρως έτοιμη, αλλά μέχρι να ζητηθούν όλα τα rixels να είναι διαθέσιμη.

4.3 Συγχρονισμός

Σε αρκετές περιπτώσεις είναι απαραίτητη η μεταφορά δεδομένων μεταξύ μπλοκς που βρίσκονται σε διαφορετικά πεδία ρολογιού. Συγκεκριμένα, τα μπλοκς Αιτήσεων Μνήμης και Εισόδου Παραθύρων βρίσκονται σε διαφορετικό ρολόι από τα υπόλοιπα.

Το μεγαλύτερο πλήθος δεδομένων που χρειάζεται να περάσει από το ένα πεδίο στο άλλο, συγχρονίζεται με φυσικό τρόπο στον Ενταμιευτή εισόδου. Η πόρτα εγγραφής συγχρονίζεται με το ρολόι της μνήμης, ενώ η πόρτα ανάγνωσης με αυτό της Μονάδας Κυρίως Επεξεργασίας. Το θέμα που απομένει εδώ είναι η συνθήκη ανταγωνισμού που δημιουργείται: πώς η Μονάδα Επεξεργασίας ξέρει πότε να διαβάσει από τον Ενταμιευτή Εισόδου.

Το πρόβλημα λύνεται με τον αυστηρό συγχρονισμό ενός μόνο bit σημαίας μεταξύ των δύο πεδίων ρολογιού. Οι μέθοδοι συγχρονισμού ενός bit στη βιβλιογραφία είναι αρκετές – πρόβλημα συναντούμε στον ταυτόχρονο συγχρονισμό αρκετών bits. Η χρήση μιας σημαίας ενός bit επιτρέπει στα δύο μπλοκς να επικοινωνούν με ένα απλό πρωτόκολλο αίτησης-απάντησης. Εδώ η αίτηση μεταφράζεται στο “υπάρχουν διαθέσιμα δεδομένα

στον ενταμιευτή” και η απάντηση “τα δεδομένα παραλήφθησαν”.

Η ίδια μέθοδος συγχρονισμού χρησιμοποιείται από όλο το κύκλωμα για την επικοινωνία μεταξύ μπλοκς που δεν είναι στο ίδιο ρολόι. Για μεταφορές δεδομένων μαζί με τις σημαίες, ο αποστολέας τοποθετεί τα δεδομένα σε έναν καταχωρητή που βρίσκεται στο ρολόι του και στέλνει τη σημαία στον παραλήπτη. Μέχρι να συγχρονιστεί η σημαία και να τη δει ο παραλήπτης, είναι σίγουρο ότι τα δεδομένα του καταχωρητή είναι ασφαλή προς ανάγνωση από το άλλο μπλοκ.

5

Μεθοδολογία και εργαλεία

Στο κεφάλαιο αυτό γίνεται μια βήμα προς βήμα περιγραφή του τρόπου με τον οποίο προσεγγίστηκε η συγκεκριμένη εργασία. Στις υποενότητες που ακολουθούν υπάρχουν πληροφορίες για τη λογική δομή των σταδίων που ακολουθήθηκαν, το χρόνο που χρειάστηκαν και τα προβλήματα που αντιμετωπίστηκαν.

5.1 Ανάλυση και προσχέδιο

Το πρώτο στάδιο της ανάπτυξης του υποσυστήματος ΑΠΚ ήταν η κατανόηση των απαιτήσεων που υπήρχαν για αυτό, η περιγραφή τους σε κείμενα και η λεπτομερειακή καταγραφή της επικείμενης υλοποίησης.

5.1.1 Κείμενο Βασικών Απαιτήσεων

Το βασικό αρχικό υλικό στο οποίο στηρίχθηκε η όλη προσπάθεια ήταν ένα κείμενο Βασικών Απαιτήσεων για το υποσύστημα. Με τη συμμετοχή σε κάποια μέρη που παρουσίαζαν ασάφειες, επιτεύχθηκε μια σταθερή πλατφόρμα εκκίνησης η οποία περιλάμβανε σημείο προς

σημείο πληροφορίες για την αλγοριθμική επεξεργασία του υποσυστήματος, το περιβάλλον του, τον τρόπο επικοινωνίας με το υπόλοιπο σύστημα, τις χρονικές και χωροταξικές απαιτήσεις του, πληροφορίες για τη διαθέσιμη τεχνολογία κλπ.

Σε αυτό το πρώτο κείμενο Βασικών Απαιτήσεων, δεν επιχειρήθηκε “εσωτερική” ανάλυση του ΑΠΚ. Ο ρόλος ενός τέτοιου κειμένου επικεντρώνεται στην σωστή γνώση του περιβάλλοντος και του κεντρικού αλγόριθμου. Πιο συγκεκριμένα, το περιβάλλον συνήθως περιγράφεται με έναν εκτενή κατάλογο κυματομορφών που σχετίζονται με τα pins εισόδου/εξόδου του υποσυστήματος που θέλουμε να περιγράψουμε και εξονυχιστική καταγραφή των διαφορετικών περιπτώσεων επικοινωνίας (αν υπάρχουν) με πίνακες και επεξήγηση σε κείμενο. Ο κεντρικός αλγόριθμος προσδιορίζεται με σαφή κεφάλαια που επεξηγούν τις επιμέρους λειτουργίες, πλήθος εικόνων που βοηθούν στην κατανόηση του κειμένου και πίνακες που συνοψίζουν τις παραμέτρους.

5.1.2 Κείμενο Περιγραφής Σχεδίασης

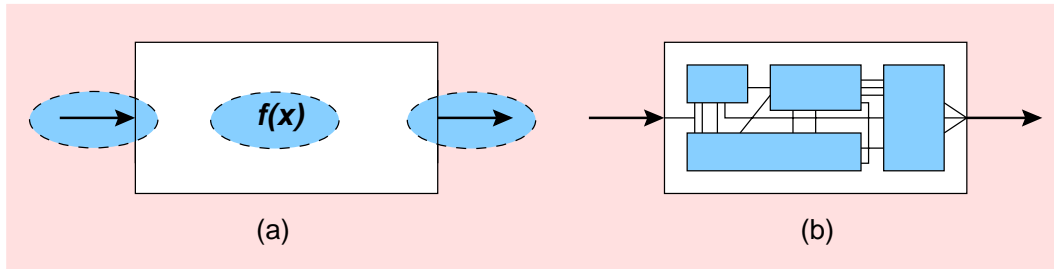
Στην επόμενη φάση της κυρίως ανάπτυξης, δημιουργήθηκε ένα δεύτερο, πιο λεπτομερειακό κείμενο που επεξηγούσε πλέον σαφώς την εσωτερική επικείμενη υλοποίηση. Αυτό το κείμενο Περιγραφής της Σχεδίασης ήταν η κύρια φάση της ανάπτυξής μας.

Ο ρόλος ενός κειμένου Περιγραφής Σχεδίασης είναι να περιγράψει στον αναγνώστη με πλήρη λεπτομέρεια το υποσύστημα, χωρισμένο στα μπλοκ που θα δημιουργηθούν στο τελική υλοποίηση πάνω στο ολοκληρωμένο κύκλωμα. Ο στόχος τέτοιων κειμένων είναι να αποτελούν αυτόνομες οντότητες περιγραφής, τέτοιες ώστε αν το συγκεκριμένο σύστημα πρέπει να αναπτυχθεί στα υπόλοιπα στάδια από κάποιον τρίτο, αυτός ο τρίτος να μπορεί να το υλοποιήσει διαβάζοντας μόνον αυτό το κείμενο.

Έτσι, το ΑΠΚ χωρίστηκε σε επιμέρους οντότητες, περιγράφηκαν όλες οι διεπαφές ανάμεσά τους σε επίπεδο pins και για κάθε μικρή οντότητα αναπτύχθηκαν κεφάλαια του κειμένου όπου περιγράφηκαν πλήρως οι επιμέρους λειτουργίες της.

Η πλήρης περιγραφή μιας μικρής οντότητας γίνεται με τρεις τρόπους: περιγραφή Μηχανών Πεπερασμένων Καταστάσεων, σχηματική περιγραφή με βασικά στοιχεία ή κείμενα και πίνακες. Και για τις τρεις περιπτώσεις, ο στόχος είναι τελικά να φαίνεται πώς θα υλοποιηθεί το κάθε ένα από τα pins εξόδου της οντότητας.

Η περιγραφή των Μηχανών Πεπερασμένων Καταστάσεων (Finite State Machines ή FSMs) σε σχήματα αποτελεί πλήρη ορισμό σχέσης σημάτων εισόδου/εξόδου, εάν όλα τα εμπλεκόμενα σήματα φαίνονται στο σχήμα με τη σωστή ονοματολογία. Αντίστοιχα, η περιγραφή σε σχήματα με βασικά στοιχεία εξυπηρετεί τον ίδιο σκοπό, εάν φαίνονται πλήρως πώς προκύπτουν τα σήματα εξόδου μέσα από δρόμους που περιλαμβάνουν “γνωστά” υλοποιή-



Σχήμα 5.1: Δομή κειμένων του υποσυστήματος

σιμους μετρητές, αθροιστές, πολυπλέκτες κλπ. Ο τρίτος τρόπος περιγραφής, απλά με κείμενα και πίνακες αποτελεί έναν λιγότερο λεπτομερειακό τρόπο εξήγησης, που όμως είναι απαραίτητος στην καταγραφή σημάτων τα οποία θα υλοποιηθούν από μεγάλα κυκλώματα συνδυαστικής λογικής – συνήθως σχετιζόμενα με πολύπλοκα σήματα ελέγχου.

Η διαφορά των δύο κειμένων στο αντικείμενο που περιγράφουν αποδίδεται στο σχήμα 5.1. Στο (a) φαίνεται πού επικεντρώνεται το κείμενο Βασικών Απαιτήσεων (επικοινωνία με άλλα υποσυστήματα και αλγοριθμική περιγραφή). Στο (b) δείχνεται ότι το κείμενο Περιγραφής Σχεδίασης ορίζει και περιγράφει επακριβώς την εσωτερική δομή του υποσυστήματος.

5.1.3 Προβλήματα και Διλήμματα

Σε όλη αυτή τη μακρά διεργασία καταμερισμού και σχεδιασμού, ένα πλήθος από προβλήματα εμφανίστηκαν και αντιμετωπίστηκαν. Πολλές φορές, η αντιμετώπισή τους δημιουργούσε σχεδιαστικά διλήμματα που είχαν διάφορες λύσεις από τη βιβλιογραφία και έπρεπε να βρεθεί η καταλληλότερη. Το δύο πιο εμφανή από τα διλήμματα που αντιμετωπίστηκαν ήταν αυτά του συγχρονισμού μεταξύ των δύο πεδίων ρολογιού και της χρονικής απαίτησης για την επεξεργασία των macroblocks.

Η λύση που περιγράφεται στο κεφάλαιο 4 για το συγχρονισμό των πεδίων του ρολογιού, η απόφαση για το ποια μπλοκς θα βρίσκονται σε ποιο πεδίο και το με ποιές ακριβώς σημαίες θα επικοινωνούν, ήταν ένα αποτέλεσμα ενδελεχούς μελέτης των διάφορων λύσεων. Η επιλογή έγινε με γνώμονα την αντοχή του κυκλώματος σε διαφορετικές συχνότητες ρολογιού, την ευκολία υλοποίησης των δρόμων αυτών στα μετέπειτα εργαλεία για την φυσική τοποθέτηση και τη διαφάνεια της λύσης για το λεπτομερή έλεγχο που θα επακολουθούσε.

Ο χρονικός περιορισμός που τίθεται στο ΑΠΚ για τη γρήγορη επεξεργασία των macroblocks, οδήγησε στην τελική λύση της συμπεριφοράς του ως μια pipeline τριών σταδίων. Και πάλι, εκτιμήθηκε ο συμβιβασμός μεταξύ της μεγαλύτερης ταχύτητας επεξεργασίας σε σχέση με τη μεγαλύτερη χωρητικότητα του κυκλώματος (χρήση των δύο ενδιάμεσων διπορτων ενταμιευτών), και αποφασίστηκε εν τέλει η συγκεκριμένη υλοποίηση.

Ένας αριθμός μικρότερων θεμάτων αντιμετωπίστηκε με παρόμοιο πνεύμα. Ο τρόπος οργάνωσης των ενταμιευτών, τα συστήματα γνώσεως ελεύθερων θέσεων, η αποδοτική συνεννόηση των τελευταίων σταδίων για τη γρήγορη έξοδο των macroblocks, οι πάμπολλοι συμβιβασμοί πολύπλοκων λύσεων έναντι απλούστερου τελικού ελέγχου είναι μερικά από αυτά, στα οποία δεν θα επεκταθούμε.

Μετά το τέλος του πολύπλοκου αυτού κειμένου Περιγραφής Σχεδίασης του ΑΠΚ, το υποσύστημα είχε καταμεριστεί σαφώς σε μικρές οντότητες, όπου η κάθε μια είχε αναλυθεί, κατανοηθεί και περιγραφεί πλήρως. Υπήρχε πλέον το κατάλληλο υπόβαθρο για την πραγματική υλοποίηση σε hardware.

5.2 Περιγραφή σε VHDL

Η σημερινή επικρατέστερη μέθοδος ανάπτυξης hardware περιλαμβάνει τη σχεδίαση των κυκλωμάτων σε κάποια ειδική γλώσσα υψηλού επιπέδου. Οι δύο δημοφιλέστερες γλώσσες είναι η Verilog και η VHDL. Η υλοποίηση του ΑΠΚ έγινε στη γλώσσα VHDL.

Τυπικά, ξεκινώντας από ένα πλήρες κείμενο Περιγραφής Σχεδίασης, η συγγραφή του κώδικα σε μια γλώσσα HDL (Hardware Description Language) όπως η VHDL είναι μια εργασία χαμηλού φόρτου, αφού τα σχεδιαστικά διλήμματα έχουν μελετηθεί πλήρως στο κείμενο. Στην προκειμένη περίπτωση, πραγματικά υπήρξε μια γρήγορη ανάπτυξη.

Ο τελικός κώδικας παρέμεινε πιστός στο κείμενο και οι οντότητες διατηρήθηκαν ακέραιες. Μια μικρή διαφοροποίηση ήταν ο επιπλέον καταμερισμός μιας μεγάλης οντότητας (της Μονάδας Κυρίως Επεξεργασίας) σε λίγες μικρότερες, καθαρά για θέματα οπτικής ευκολίας στην ανάγνωση της VHDL.

5.3 Έλεγχος και ανάπτυξη εργαλείου

Ο μεγαλύτερος χρόνος ανάπτυξης ενός κυκλώματος απαιτείται για την περίοδο που το κύκλωμα ελέγχεται για τη σωστή του λειτουργία. Στην προκειμένη περίπτωση, το ΑΠΚ δεν εξετάστηκε μεν ακόμα εξονυχιστικά, αλλά δημιουργήθηκε η απαραίτητη πλατφόρμα και το σύστημα άρχισε να δουλεύει.

Πιο συγκεκριμένα, η διαδικασία του ελέγχου ξεκίνησε με την ανάπτυξη ενός καινούριου εργαλείου για έλεγχο σε VHDL με έναν ειδικό, πιο εύχρηστο από τον κανονικό, τρόπο. Τα εργαλεία που χρησιμοποιήθηκαν για την προσομοίωση της VHDL ανήκουν στο πακέτο NCSim της Cadence. Τυπικά, γράφεται κώδικας VHDL που προσομοιώνει το περιβάλλον

του υποσυστήματος (Testbench) ο οποίος προσομοιώνεται μαζί με αυτό. Ένα από τα πιο χρονοβόρα στάδια του ελέγχου είναι αυτό της εύρεσης των πρώτων χονδροειδών λαθών. Εκεί, ο κώδικας ελέγχου τροποποιείται αρκετά συχνά με σκοπό να τα εντοπίσει, προκαλώντας μεγάλη χρονική καθυστέρηση εξ' αιτίας της συνεχούς εκ νέου μεταγλώττισης.

Το καινούριο εργαλείο που αναπτύχθηκε ειδικά για αυτήν τη φάση, είναι ένα γενικό Testbench σε VHDL το οποίο μεταγλωττίζεται μόνον μια φορά στο σύστημα. Κατόπιν, διαβάζει μια εύχρηστη απλοποιημένη γλώσσα – παρόμοια της VHDL και ειδικά φτιαγμένη για έλεγχο – από απλά αρχεία κειμένου, και την εκτελεί. Το εργαλείο αυτό λειτουργεί σαν μεταφραστής (interpreter) πετυχαίνοντας διπλό σκοπό: τη μείωση του χρόνου ελέγχου λόγω συνεχών μεταγλωττίσεων και την απλοποίηση της συγγραφής του Testbench, αφού η καινούρια ψευδογλώσσα παρέχει ειδικά απλοποιημένες δομές για πιο γρήγορο έλεγχο.

Το εργαλείο αναπτύχθηκε και ελέγχθηκε πλήρως, και η σταθερή του έκδοση σήμερα χρησιμοποιείται σε πολλά διαφορετικά υποσυστήματα εκτός του ΑΠΚ. Για το ΑΠΚ, χρησιμοποιήθηκαν τρεις ανεξάρτητες οντότητες του εργαλείου αυτού, που η κάθε μια από αυτές προσομοίωνε με απλοϊκό τρόπο τα τρία άλλα υποσυστήματα του περιβάλλοντος.

Ο έλεγχος που πραγματοποιήθηκε στο ΑΠΚ απέδειξε τη λειτουργία του υποσυστήματος για μερικές βασικές περιπτώσεις. Αυτό που υπολείπεται είναι να γραφεί ένα μοντέλο αναφοράς σε software, και να συγκριθούν τα αποτελέσματά του με αυτά του ΑΠΚ για όλες τις πιθανές περιπτώσεις λειτουργίας.

5.4 Σύνθεση στα 0.18 μm και Χρονική Ανάλυση

Στη συνέχεια, το υποσύστημα πέρασε από τη φυσική ροή της υλοποίησης σε hardware και δημιουργήθηκε έτσι ένας πραγματικός πυρήνας πυριτίου (silicon core) για αυτό. Το πρώτο βασικό στάδιο ήταν η αναπαράσταση της γλώσσας υψηλού επιπέδου σε standard cells της τεχνολογίας που χρησιμοποιήσαμε.

Η σύνθεση του κώδικα VHDL έγινε με το εργαλείο Design Compiler της Synopsys. Η τεχνολογία που ήταν διαθέσιμη ήταν στα 0.18 μm της Virtual Silicon Technology.

Κατά τη διαδικασία της σύνθεσης, η κάθε βασική οντότητα από τις 5 του ΑΠΚ αντιμετωπίστηκε ξεχωριστά. Για κάθε μια ορίστηκαν οι παράμετροι λειτουργίας (συχνότητα ρολογιού, επιθυμητός χώρος σε τετραγωνικά χιλιοστά επιφάνειας πυριτίου) και άρχισε ένας κύκλος βελτιστοποίησης των αποτελεσμάτων.

Μια σημαντική λεπτομέρεια του σχεδιασμού ήταν ότι κάθε οντότητα είχε ένα ακριβώς ρολόι. Οι δρόμοι επικοινωνίας μεταξύ δύο μπλοκς με διαφορετικά ρολόγια ήταν έτσι ευδιάκριτοι στο κορυφαίο επίπεδο του ΑΠΚ (top level), πράγμα που δεν δημιούργησε προβλή-

ματα στη σύνθεση: η σύνθεση έγινε για κάθε μπλοκ αυτόνομα. Οι συγχρονιστές συνθέθηκαν επίσης αυτόνομα, υποδεικνύοντας στο εργαλείο ποιο ακριβώς μονοπάτι του ενός bit περνάει από το ένα ρολόι στο άλλο, έτσι ώστε να μην προσπαθήσει να το βελτιστοποιήσει χρονικά. Η διαδικασία αυτή γίνεται με την υπόδειξη για εσφαλμένα χρονικά μονοπάτια (false paths).

Το αποτέλεσμα της σύνθεσης από το Design Compiler ήταν μια περιγραφή των standard cells της τεχνολογίας στις 5 ανεξάρτητες οντότητες, και η διασύνδεση στο κορυφαίο επίπεδο των οντοτήτων, των συγχρονιστών και των μνημών με καλώδια. Η περιγραφή αυτή δημιουργήθηκε αυτόματα από το εργαλείο και ήταν στη γλώσσα Verilog (Verilog netlist).

Κατόπιν, έγινε η Χρονική Ανάλυση του Verilog netlist χρησιμοποιώντας το εργαλείο Pearl της Cadence. Εκεί, εισάγοντας τις συχνότητες των δύο ρολογιών ελέγχθηκε ότι το κύκλωμα όντως ανταποκρίνεται στις χρονικές προδιαγραφές του, για τη συγκεκριμένη υλοποίηση στη συγκεκριμένη τεχνολογία.

Η κανονική διαδικασία περιλαμβάνει ένα βρόγχο στο σημείο αυτό. Ο χρονικός αναλυτής συνήθως ανακαλύπτει παραβιάσεις για την τελική συχνότητα ρολογιού και ο σχεδιαστής περιγράφει και πάλι κάποια κομμάτια του κυκλώματος. Στο ΑΠΚ δεν υπήρξε κάτι τέτοιο, αφού η σχεδίαση ήταν αναλυτική από τα πρώτα στάδια (κείμενο περιγραφής σχεδίασης). Ειδικά στα κομμάτια που υπήρχε αρκετή λογική έγινε μια εκτίμηση με κομμάτια της βιβλιοθήκης δομικών στοιχείων για τα 0.18 μm, όπου επιβεβαιωνόταν εάν το κύκλωμα “χωράει” άνετα σε μια περίοδο ρολογιού.

Ο χρονικός αναλυτής χρειάστηκε μεγαλύτερη λεπτομέρεια στο χειρισμό των δρόμων μεταξύ μπλοκ διαφορετικού ρολογιού. Εκτός από την υπόδειξη των συγχρονιστών ως εσφαλμένων χρονικών μονοπατιών, οι καταχωρητές και οι μνήμες όπου γράφονταν με το ένα ρολόι αλλά διαβάζονταν με το άλλο έπρεπε επίσης να δηλωθούν μεμονωμένα ως τέτοια μονοπάτια. Στην πράξη, πρόκειται για εξαιρέσεις των συγκεκριμένων κομματιών από τη χρονική ανάλυση, οι οποίες είμαστε σίγουροι ότι θα δουλέψουν λόγω της φύσεως της σχεδίασης (τα δεδομένα διαβάζονται κατόπιν συνεννόησης των δύο μπλοκ μέσω των σημάτων που ανταλλάσσουν).

Έτσι, το εργαλείο της Χρονικής Ανάλυσης επιβεβαίωσε ότι το συγκεκριμένο κύκλωμα πληρεί τις προϋποθέσεις για τα δύο ρολόγια. Πλέον αυτό που έμενε ήταν η φυσική τοποθέτηση των standard cells της τεχνολογίας στο χώρο, η διασύνδεσή τους με μέταλλα και η εκ νέου ανάλυση του κυκλώματος για την επιβεβαίωσή του.

5.5 Τοποθέτηση και διασύνδεση

Τα εργαλεία που χρησιμοποιήθηκαν για το τελευταίο αυτό κομμάτι της φυσικής ροής, ήταν το Silicon Ensemble και το Design Planner της Cadence. Ακολουθήθηκε ιεραρχικός τρόπος τοποθέτησης και διασύνδεσης.

Οι ανεξάρτητες οντότητες τοποθετήθηκαν και διασυνδέθηκαν η καθε μια ξεχωριστά στο Silicon Ensemble. Τα στάδια αυτά περιλάμβαναν για την κάθε μια τη δημιουργία ενός μπλοκ πυριτίου, τη χάραξη δακτυλίων ρεύματος, την τοποθέτηση των standard cells από το Verilog netlist, τη δημιουργία δέντρου ρολογιού (clock tree), τη διασύνδεση του ρολογιού, τη διασύνδεση των οδών ρεύματος και τη διασύνδεση όλων των μετάλλων επικοινωνίας των standard cells. Σε πολλά σημεία ενδιαμέσως πραγματοποιήθηκε εκ νέου χρονική ανάλυση, έτσι ώστε να επιβεβαιώνεται ότι τα συγκεκριμένα στάδια είναι όντως εντός των απαιτήσεων.

Μετά από αυτήν τη φάση του Silicon Ensemble, υπήρχαν πλέον όλες οι απαραίτητες οντότητες πυριτίου για να γίνει η τελική κάτοψη. Χρησιμοποιώντας το Design Planner και στη συνέχεια το Silicon Ensemble τα μπλοκς που είχαν δημιουργηθεί και οι δύο μνήμες τοποθετήθηκαν στο τελικό κομμάτι πυριτίου και διασυνδέθηκαν. Μια καινούρια χρονική ανάλυση για όλο το υποσύστημα επιβεβαίωσε και πάλι τη σωστή λειτουργία.

Τέλος, εξάγοντας από τα εργαλεία μια καινούρια περιγραφή σε Verilog για όλον τον πυρήνα που δημιουργήσαμε, η οποία περιλάμβανε και όλες τις πληροφορίες για καθυστερήσεις λόγω μετάλλων και παρασιτικών χωρητικότητων, πραγματοποιήθηκε μια προσωμίωση του όλου κυκλώματος για μερικές βασικές περιπτώσεις ώστε να επιβεβαιωθεί ότι τα αποτελέσματα του πυριτίου δεν είχαν κάποιο σφάλμα στη διαδικασία της φυσικής ροής.

6

Αποτελέσματα

Στον πίνακα 6.1 φαίνονται μερικά τεχνικά στοιχεία της τελικής υλοποίησης του ΑΠΚ. Το τελικό αποτέλεσμα βρέθηκε σύμφωνο με τις αρχικές προδιαγραφές, τόσο σε επίπεδο συχνότητας ρολογιών όσο και χώρου πάνω στο ολοκληρωμένο κύκλωμα.

Σύμφωνα με τις προδιαγραφές αυτές, το σύστημα του αποκωδικοποιητή μπορεί να πετύχει την ταυτόχρονη αποκωδικοποίηση δύο Video Elementary Streams. Το ΑΠΚ σε αυτήν την περίπτωση δουλεύει τμηματικά τις δύο εικόνες – macroblock της μίας εναλλάξ με macroblock της άλλης.

Στον πίνακα 6.2 υπάρχει ένα βασικό χρονοδιάγραμμα των φάσεων της ανάπτυξης. Το συνολικό διάστημα του ενός ανθρωποχρόνου δαπανήθηκε για το ΑΠΚ.

Στον πίνακα 6.3 φαίνονται τα εργαλεία που μαθεύτηκαν και χρησιμοποιήθηκαν για την ανάπτυξη αυτή.

Τέλος, στο σχήμα 6.1 διακρίνεται η πορεία ενός από τα μπλοκς κατά την τοποθέτηση και τη διασύνδεση. Από αριστερά προς τα δεξιά και από πάνω προς τα κάτω φαίνονται:

- ☞ Το αρχικό περίγραμμα του μπλοκ, με τοποθετημένα τα σημεία εισόδου εξόδου και τις γραμμές για τα standard cells.

| | |
|-------------------------------------|--|
| Κείμενο Βασικών Απαιτήσεων | 36 σελίδες |
| Κείμενο Περιγραφής Σχεδίασης | 77 σελίδες |
| Κώδικας ΑΠΚ | VHDL, 4398 γραμμές |
| Κώδικας ελέγχου | VHDL, 2650 γραμμές |
| Μπλοκς | 5 κύριας λογικής, 7 συγχρονιστές, 5 μνήμες |
| Μνήμες | 1 x 256 x 16, 1 πόρτα ανάγνωσης, 1 εγγραφή |
| | 4 x 256 x 32, 1 πόρτα ανάγνωσης, 1 εγγραφή |
| Συχνότητες ρολογιών | 100 MHz και 75 MHz |
| Καθαρό εμβαδό λογικής | 206,190 μm^2 |
| Καθαρό εμβαδό μνημών | 781,722 μm^2 |
| Στοιχεία τεχνολογίας | 0.18 μm , 6 μέταλλα, 1.8 Volt |

Πίνακας 6.1: Στοιχεία του ΑΠΚ

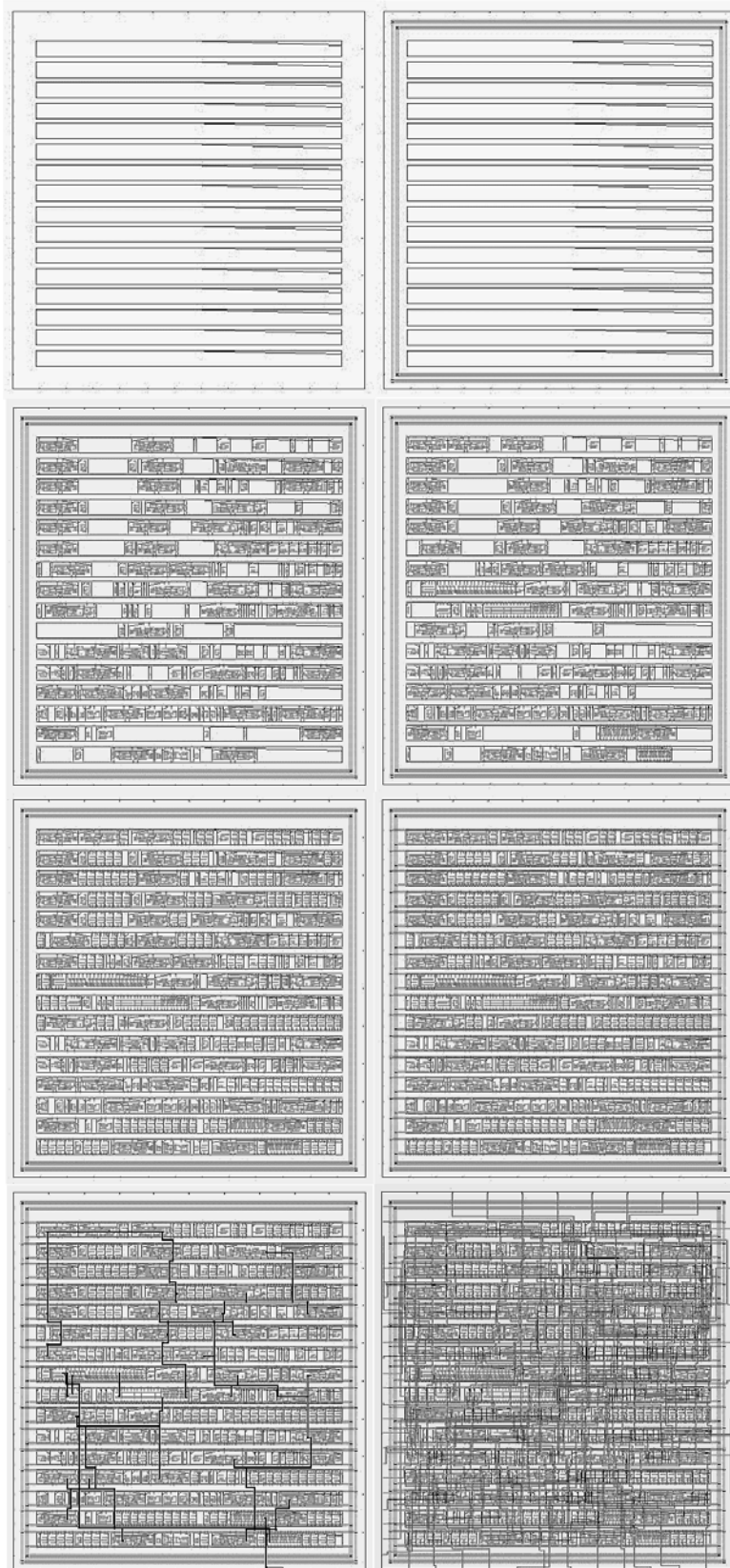
| | |
|--|-----------------|
| Συμμετοχή στο κείμενο Βασικών Απαιτήσεων | 1 μήνας |
| Συγγραφή κειμένου Περιγραφής Σχεδίασης | 4 μήνες |
| Ανάπτυξη κώδικα VHDL | 1 μήνας |
| Ανάπτυξη εργαλείου ελέγχου | 2 μήνες |
| Έλεγχος | 1 μήνας |
| Σύνθεση και Χρονική Ανάλυση | 1 μήνας |
| Τοποθέτηση και Διασύνδεση | 1 μήνας |
| Σύνολο | 12 μήνες |

Πίνακας 6.2: Χρόνος για την ανάπτυξη του ΑΠΚ

- ⇒ Τοποθέτηση των δύο δακτυλίων τροφοδοσίας σε μέταλλο-1 και μέταλλο-2.
- ⇒ Τοποθέτηση των standard cells που έχουν προκύψει από τη σύνθεση, σύμφωνα με τους χρονικούς περιορισμούς για το μπλοκ.
- ⇒ Υπολογισμός και τοποθέτηση των ενισχυτών για το δέντρο ρολογιού.
- ⇒ Τοποθέτηση “filler cells” (cells που γεμίζουν τα κενά στις σειρές για να μην παρατηρείται latchup στο τελικό κύκλωμα).
- ⇒ Σύνδεση των γραμμών με τους δακτυλίους τροφοδοσίας (οριζόντια σε μέταλλο-1).
- ⇒ Σύνδεση του δέντρου ρολογιού με όλα τα flip-flops και τους ενισχυτές.
- ⇒ Διασύνδεση όλων των υπόλοιπων δρόμων μεταξύ των standard cells και των εισόδων/εξόδων του μπλοκ.

| | Εργαλείο | Εταιρεία |
|-------------------------------------|----------------------|-----------------|
| Προσομοίωση Verilog και VHDL | Affirma NCSim | Cadence |
| Σύνθεση | Design Compiler | Synopsys |
| Χρονική Ανάλυση | Pearl | Cadence |
| Τοποθέτηση και Διασύνδεση | Silicon Ensemble | Cadence |
| Ιεραρχική κάτοψη | Design Planner | Cadence |
| Διασύνδεση σε top-level | Chip Assembly Router | Cadence |

Πίνακας 6.3: Εργαλεία που χρησιμοποιήθηκαν



Σχήμα 6.1: Τοποθέτηση και διασύνδεση ενός μπλοκ

7

Επίλογος

7.1 Εμπειρία

Κατά τη διάρκεια της μεταπτυχιακής αυτής εργασίας στο ΑΠΚ, αποκτήθηκε πολύτιμη εμπειρία σε πολλά πεδία. Η ολοκληρωμένη δουλειά σε ένα πραγματικό σχέδιο συνέβαλε στην κατανόηση και την αντιμετώπιση των προβλημάτων και των κινδύνων που μπορεί να ανακύψουν.

Πιο συγκεκριμένα, ήταν η πρώτη φορά που απαιτήθηκε η συγγραφή κειμένων τέτοιας περιεκτικότητας, λεπτομέρειας και αυστηρής δομής. Η δυσκολία που αντιμετωπίζουν οι περισσότεροι φοιτητές στη συστηματική γραπτή τεκμηρίωση των απόψεών τους, όταν πρόκειται για μεγάλο όγκο δουλειάς, έπρεπε να ξεπεραστεί. Τα αυστηρά πλαίσια που τίθενται για τέτοιου είδους κείμενα (συγκεκριμένες παράγραφοι, μεγάλο πλήθος πινάκων και εικόνων, κλπ) είναι μεν περιοριστικά αλλά καθοδηγούν τελικά προς το επιθυμητό αποτέλεσμα το οποίο δεν είναι άλλο από ένα τελικό κείμενο που να διαβάζεται εύκολα από κάποιον τρίτο.

Η γνώση της γλώσσας VHDL ήταν κάτι καινούριο που αποκτήθηκε, αφού μέχρι πρότινος χρησιμοποιούσαμε μόνο τη γλώσσα Verilog. Η VHDL γενικά εμπεριέχει πιο αυστηρή σύ-

νταξη και πιο στρυφνό τρόπο περιγραφής, αλλά τελικά καταλήγει να δεσμεύει το συγγραφέα να είναι πιο ξεκάθαρος στα κομμάτια λογικής που περιγράφει.

Η όλη φιλοσοφία του εργαλείου ελέγχου που αναπτύχθηκε ενδιάμεσως προσέφερε μια εντελώς καινούρια πνοή. Η υλοποίηση ενός ενδιάμεσου μεταφραστή σε VHDL ήταν από μόνη της μια καρποφόρα διαδικασία, αλλά η όλη σύλληψη και το τελικό αποτέλεσμα απέδωσε μια ακόμα πιο ολοκληρωμένη εικόνα για τα περιβάλλοντα ελέγχου.

Τα εργαλεία που χρησιμοποιήθηκαν ήταν και πάλι κάτι το καινούριο. Ο μεν μεταγλωττιστής και προσομοιωτής για τη VHDL διέφερε σημειακά από παλαιότερους που είχαμε χρησιμοποιήσει, αλλά τα εργαλεία της σύνθεσης και της φυσικής ροής ήταν κάτι εντελώς πρωτόγνωρο. Πρέπει να σημειωθεί ότι μόνο ένα μικρό τμήμα τους φάνηκε στη χρήση που τους έγινε, διότι παρέχεται ένας τεράστιος αριθμός παραμέτρων που απευθύνεται τελικά σε όλη την πιθανή γκάμα επαγγελματικών προδιαγραφών. Πάντως, αποκτήθηκε αρκετή οικειότητα με τα εργαλεία αυτά ώστε να αποτελούν μια σταθερή αφετηρία για μελλοντική επέκταση σε εξειδικευμένα θέματα που δεν συναντήθηκαν εδώ.

7.2 Μελλοντική εργασία

Στην εργασία για το υποσύστημα ΑΠΚ παραμένει η εκκρεμότητα του λεπτομερούς αλγοριθμικού ελέγχου του. Για το σκοπό αυτό, πρέπει να αναπτυχθεί ένα ανεξάρτητο μοντέλο σε software, π.χ. σε γλώσσα C, το οποίο να δέχεται ως είσοδο ένα στοιχειώδες ρεύμα video και να παράγει σαν αποτέλεσμα ένα αρχείο εξόδου που περιέχει την πρόβλεψη κίνησης.

Με τη χρήση αυτού του μοντέλου, θα πρέπει να αναπτυχθεί ένα πληρέστερο Testbench το οποίο θα τρέχει αυτόματα το μοντέλο και μια προσομοίωση του ΑΠΚ και θα συγκρίνει τα αποτελέσματά τους. Η αλγοριθμική επιβεβαίωση θα είναι πλήρης μόνον όταν αυτό επιτύχει για όλες τις παραλλαγές των τρόπων λειτουργίας του ΑΠΚ.

Σε άλλες μελλοντικές επεκτάσεις, θα ήταν χρήσιμο να μελετηθούν ακόμα πιο επιθετικοί τρόποι χρήσης της φυσικής ροής, ώστε το τελικό κομμάτι πυριτίου να είναι μικρότερο, να χρειάζεται λιγότερη ισχύ λειτουργίας ή να μπορεί να λειτουργεί και για πιο γρήγορες συχνότητες ρολογιών.

8

Αναφορές

- [1] IEEE Standard: "Information technology - Generic coding of moving pictures and associated audio information: Video", ISO/IEC 13818-2:1996(E).
- [2] IEEE Standard: "Information technology - Generic coding of moving pictures and associated audio information: Systems", ISO/IEC 13818-1:1996(E).
- [3] Tektronix, "A Guide to MPEG Fundamentals and Protocol Analysis (Including DVB and ATSC)".
http://www.tektronix.com/Measurement/App_Notes/mpegfund/25W_11418_3.pdf
- [4] John Watkinson, "MPEG-2", Butterworth-Heinemann, March 1998.
- [5] Peter Symes, "Video Compression", McGraw-Hill, 1998.
- [6] "MPEG Starting Points and FAQs",
<http://www.mpeg.org/MPEG/starting-points.html>
- [7] Leonardo Chiariglione - Convenor, "A Short MPEG-2 description",
<http://mpeg.telecomitalia.com/standards/mpeg-2/mpeg-2.htm>