



Πανεπιστήμιο Πειραιώς  
Σχολή Τεχνολογιών Πληροφορικής και Επικοινωνιών  
Τμήμα Ψηφιακών Συστημάτων  
Επικοινωνίες Πολυμέσων

## Επεξεργασία Πολυμέσων σε Ενσωματωμένα Συστήματα

Μελέτη και Σύγκριση Ενσωματωμένων  
Συστημάτων Επεξεργασίας Πολυμέσικών  
Δεδομένων

Θεοφάνης Τριανταφύλλης  
Ε17150  
fanis\_30fillis@outlook.com

Επιβλέπων Καθηγητής:  
Ηλίας Μαγκλογιάννης, Καθηγητής

ΠΕΙΡΑΙΑΣ  
18/06/2021

Η εικόνα 3.1 είναι από το Warehouse-Scale Video Acceleration: Co-design and Deployment in the Wild

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8317-2/21/04.

<https://doi.org/10.1145/3445814.3446723>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

# Περιεχόμενα

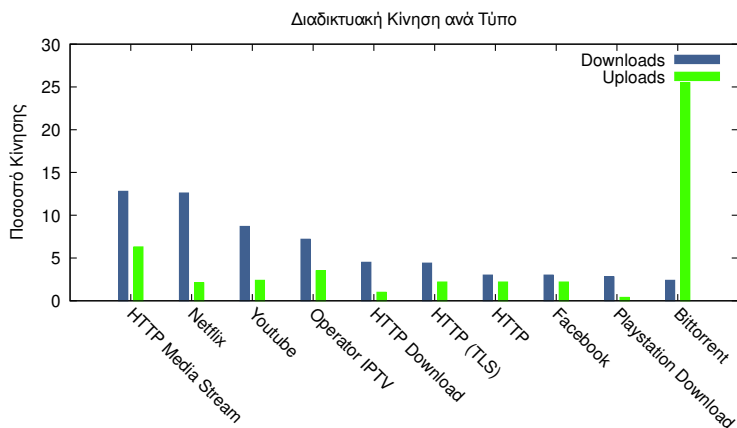
<b>0</b>	<b>Εισαγωγή</b>	<b>1</b>
<b>1</b>	<b>Συστήματα Επεξεργασίας Πολυμέσων</b>	<b>3</b>
1.1	Είσοδος και Έξοδος Δεδομένων	3
1.2	Επεξεργαστές των Δεδομένων	3
1.2.1	Application Specific Integrated Circuits (ASIC) και Application Specific Standard Product (ASSP)	3
1.2.2	Field Programmable Gate Arrays (FPGA)	4
1.2.3	Μικροεπεξεργαστές	4
1.2.4	Κατανάλωση Ισχύος	5
1.3	Μνήμες	5
1.3.1	Ιεραρχία Μνήμης	5
1.3.2	Μνήμες Τυχαίας Προσπέλασης - Random Access Memory (RAM)	6
<b>2</b>	<b>Συστήματα για Κωδικοποίηση Video</b>	<b>7</b>
2.1	Κωδικοποίηση JPEG 2000	7
2.1.1	Έννοιες	7
2.1.2	Προ επεξεργασία	7
2.1.3	Κωδικοποίηση	8
2.2	Κωδικοποίηση με Παράλληλο EBCOT	8
2.2.1	Κωδικοποίηση Επιπέδου Bit	8
2.2.2	Συμπίεση	9
2.2.3	Αρχιτεκτονική	9
2.2.4	Απόδοση	11
2.3	Σύστημα για Κωδικοποίηση Βίντεο με Χρήση JPEG2000	11
2.3.1	Υλοποίηση του Μετασχηματισμού	12
2.3.2	Υλοποίηση Υλικού	14
2.3.3	Απόδοση Συστημάτων	15
2.4	Σύγκριση Συστημάτων	17
2.4.1	Αρχιτεκτονική	17
2.4.2	Απόδοση	17
<b>3</b>	<b>Κωδικοποίηση Βίντεο για το Youtube</b>	<b>19</b>
3.1	Σύστημα Επεξεργασίας	19
3.1.1	Ολικός Σχεδιασμός Συστήματος	20
3.1.2	Κεντρικός Σχεδιασμός Κωδικοποιητή VCU	20
3.1.3	Σχεδιασμός Λογισμικού και Υλικού	21
3.1.4	Σύνθεση Υψηλού Επιπέδου	22
3.2	Απόδοση	22
3.3	Συμπέρασμα	24

# Κεφάλαιο 0

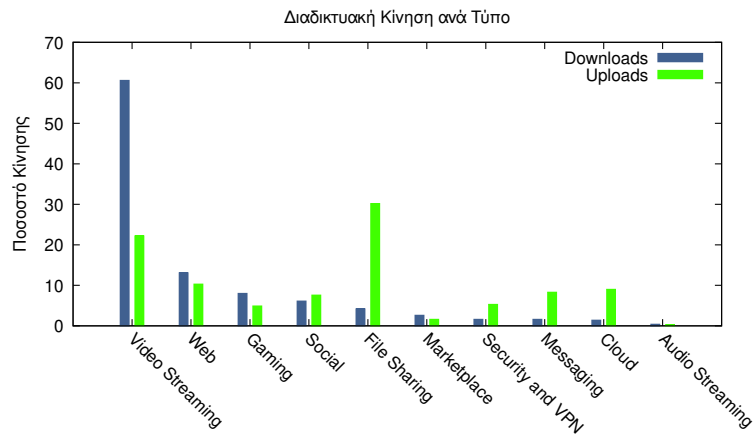
## Εισαγωγή

Τα τελευταία χρόνια έχει παρατηρηθεί η άνοδος ενσωματωμένων συστημάτων που παίζουν όλο και μεγαλύτερο ρόλο στην καθημερινότητα των ανθρώπων, όπως οι συσκευές IoT και λοιπές έξυπνες συσκευές οι οποίες επεξεργάζονται πολυμεσικά δεδομένα. Γι' αυτό το λόγο είναι απαραίτητη η αποδοτική επεξεργασία πολυμεσικών δεδομένων σε συστήματα χαμηλής απόδοσης. Επίσης πλέον δεν στέκει ο νόμος του Moore, δηλαδή η απόδοση των επεξεργαστών έχει σταθεροποιηθεί [1], αυτό σημαίνει ότι για να επιτευχθούν καλύτερες αποδόσεις πρέπει να υποστηθούν ειδικευμένες αρχιτεκτονικές ανάλογα με το πρόβλημα.

Το 2019 το 60% της συνολικής κίνησης ήταν για την μετάδοση Video, όσο αφορά τις εφαρμογές το 12.8 % ανήκε σε μετάδοση πολυμέσων μέσω του HTTP, το 12.6 % στο Netflix και το 8.7% στο Youtube, όπως φαίνεται στην εικόνα 1 [2]. Επίσης τεράστιο κομμάτι της διαδικτυακής κίνησης παίρνουν τα πολυμεσικά δεδομένα, όπως φαίνεται στην εικόνα 2. Είναι ξεκάθαρο ότι η μετάδοση πολυμέσων είναι ένα τεράστιο κομμάτι της κίνησης και της επεξεργαστικής ισχύς που καταναλώνεται κάθε χρόνο.



Εικόνα 1: Δεδομένα Εφαρμογών



Εικόνα 2: Δεδομένα Κίνησης Διαδικτύου

Αυτή η εργασία έχει ως σκοπό να περιγράψει την διαδικασίες επεξεργασίας πολυμέσων σε ενσωματωμένα συστήματα, δηλαδή για υλοποιήσεις αλγορίθμων σε υλικό καθώς και υλοποιήσεις σε λογισμικό και υλικό.

Το πρώτο κεφάλαιο αφορά το υπόβαθρο των ενσωματωμένων συστημάτων και των αρχιτεκτονικών υπολογιστών.

Το δεύτερο κεφάλαιο αναλύει δύο υλοποιήσεις του αλγορίθμου JPEG 2000 σε ενσωματωμένα συστήματα και τις συγκρίνει.

Το τρίτο κεφάλαιο αναλύει την αρχιτεκτονική που σχεδίασε η Google για τις αυξημένες απαιτήσεις κωδικοποίησης βίντεο του Youtube και σε άλλες υπηρεσίες που φροσφαίρει.

# Κεφάλαιο 1

## Συστήματα Επεξεργασίας Πολυμέσων

Ένα τυπικό σύστημα επεξεργασίας δεδομένων αποτελείται από τον επεξεργαστή, τα υποσυστήματα εισόδου και εξόδου καθώς και την μνήμη. Το καθένα από αυτά είναι κρίσιμο για την απόδοση του συστήματος.

### 1.1 Είσοδος και Έξοδος Δεδομένων

Οι συσκευές εισόδου και εξόδου είναι απαραίτητες στα συστήματα επεξεργασίας πολυμέσων, η είσοδος και η έξοδος μπορεί να είναι μέρος του συστήματος ή και όχι, δηλαδή να προέρχονται από το εξωτερικό του συστήματος.

Μερικές συσκευές που χρησιμοποιούνται για είσοδο και έξοδο είναι:

- Οθόνες
- Οθόνες Αφής
- Πληκτρολόγια
- Αισθητήρες

### 1.2 Επεξεργαστές των Δεδομένων

Ο επεξεργαστής ενός συστήματος είναι το πιο σημαντικό κομμάτι όσο αφορά την απόδοση του, άρα το κάθε σύστημα πρέπει να έχει τον κατάλληλο επεξεργαστή ώστε να κάνει αποδοτικά την επεξεργασία που πρέπει να κάνει. Υπάρχουν πολλοί τύποι επεξεργαστών που μπορούν να επιλεγθούν από έναν σχεδιαστή ενσωματωμένων συστημάτων.

#### 1.2.1 Application Specific Integrated Circuits (ASIC) και Application Specific Standard Product (ASSP)

Τα ASIC είναι ολοκληρωμένα τα οποία έχουν σχεδιαστεί αποκλειστικά για μια λειτουργία, είναι αποδοτικά και έχουν χαμηλή κατανάλωση ενέργειας, όμως είναι πολύ ακριβά στον σχεδιασμό και στην υλοποίηση [3].

Τα ASSP είναι και αυτά σχεδιασμένα αποκλειστικά για μια λειτουργία αλλά σε αντίθεση με τα ASIC σχεδιάζονται από μια εταιρία που τα πουλάει σε πολλούς πελάτες, ενώ τα ASIC συνήθως σχεδιάζονται για τις ανάγκες μιας εταιρίας.

### 1.2.2 Field Programmable Gate Arrays (FPGA)

Τα FPGA είναι λογικές συσκευές που επιτρέπουν την κατασκευή της λειτουργίας που επιθυμείται σε έναν κύκλωμα χωρίς να χρειάζεται να κατασκευαστεί ένα ειδικευμένο κύκλωμα. Αυτά τα ολοκληρωμένα έχουν πύλες τις οποίες ο προγραμματιστής συνδέει ώστε να σχεδιάσει το επιθυμητό κύκλωμα. Έχουν το πλεονέκτημα ότι τα κυκλώματα τους μπορούν να αλλάξουν μετά από τον προγραμματισμό, σε αντίθεση με τα ASIC/ASSP κυκλώματα. Είναι πιο αποδοτικά από τους επεξεργαστές αλλά είναι απαιτητικά στην ενέργεια[4, 5].

### 1.2.3 Μικροεπεξεργαστές

Οι μικροεπεξεργαστές είναι πολύ ευέλικτοι και μπορούν να πραγματοποιήσουν οποιαδήποτε επεξεργασία χρειάζεται. Μια από τις απαιτήσεις στα ενσωματωμένα συστήματα είναι οι πόροι, συνήθως οι επεξεργαστές αυτοί έχουν διαφορετικές αρχιτεκτονικές από τους επεξεργαστές που χρησιμοποιούνται σε προσωπικούς υπολογιστές, αυτό σημαίνει ότι μπορεί να έχει μικρότερο σύνολο εντολών και κατά συνέπεια να είναι πιο απλό και πιο αποδοτικά ως προς την κατανάλωση ενέργειας.

#### Σύνολα Εντολών - Instruction Sets

Τα σύνολα εντολών αποτελούνται από τις εντολές που μπορεί να εκτελέσει ο επεξεργαστής, υπάρχουν δύο είδη συνόλων εντολών:

- RISC - Reduced Instruction Set Computer  
Είναι ένα πιο απλό σύνολο εντολών με λίγες εντολές, έχει χαμηλή κατανάλωση ενέργειας
- CISC - Complex Instruction Set Computer  
Είναι πολύπλοκο σύνολο εντολών, είναι πιο γρήγορο αλλά είναι πιο απαιτητικό ως προς την ενέργεια που χρειάζεται για να λειτουργήσει

Μερικά σύνολα εντολών μπορούν να έχουν ειδικές εντολές για πολυμεσικές διεργασίες.

#### Είδη Επεξεργαστών

Οι επεξεργαστές μπορούν να χωριστούν σε υποκατηγορίες:

- Very Long Instruction Word (VLIW)  
Είναι επεξεργαστές που χρησιμοποιούνται για υπολογιστικά ακριβές διαδικασίες. Μπορούν να εκτελέσουν ορισμένες εντολές παράλληλα, κατά τη διάρκεια της μετάφρασης σε κώδικα μηχανής αναγνωρίζονται οι εντολές που μπορούν να εκτελεστούν παράλληλα, τις βάζει σε ένα πακέτο εντολών και τις εκτελεί παράλληλα κατά τη διάρκεια της εκτέλεσης του προγράμματος [3].
- Επεξεργαστές Πολλαπλών Πυρήνων  
Η δύναμη ενός μονοπύρηνου επεξεργαστή μπορεί να αυξηθεί μέχρι ενός σημείου καθώς υπάρχει περιορισμός στην ενέργεια που μπορεί να καταναλώσει ο επεξεργαστής [3], γι' αυτό υπάρχουν επεξεργαστές με πολλούς πυρήνες. Υπάρχουν ομογενείς επεξεργαστές, όπου όλοι οι πυρήνες είναι του ίδιου τύπου καθώς και ανομοιογενείς επεξεργαστές, όπου μπορούν να υπάρχουν διαφορετικοί πυρήνες για διαφορετικούς σκοπούς[3].
- Graphics Processing Units (GPU)  
Αυτοί οι επεξεργαστές προορίζονται για επεξεργασία γραφικών, οι σημερινοί επεξεργαστές εκτελούν πολλές διεργασίες ταυτόχρονα [3].

- Multiprocessor Systems on a Chip (MPSoCs)  
Σε αυτά τα συστήματα έχουμε πολλές ανομοιογενείς λειτουργίες σε ένα ολοκληρωμένο. Με αυτή την αρχιτεκτονική μπορούμε να επιτύχουμε απόδοση ισχύος κοντά στα ολοκληρωμένα ASIC.
- Digital Signal Processors  
Αυτοί οι επεξεργαστές είναι ειδικά σχεδιασμένοι για την επεξεργασία ψηφιακών συστημάτων.

### 1.2.4 Κατανάλωση Ισχύος

Η κατανάλωση ενέργειας είναι σε ορισμένες περιπτώσεις όσο σημαντικός και ο χρόνος εκτέλεσης [6]. Η ισχύς ορίζεται ως η κατανάλωση ενέργειας ανά μια μονάδα χρόνου. Η ισχύς που απαιτείται καθορίζει το μέγεθος του τροφοδοτικού, τον σχεδιασμό και την ψύξη του συστήματος [3]. Η ανάγκη για την ελαχιστοποίηση της απαιτούμενης ενέργειας είναι μεγάλη στις εφαρμογές κινητών, καθώς οι μπαταρίες βελτιώνονται με αργούς ρυθμούς [3]. Επίσης σε ορισμένες περιπτώσεις μπορεί να έχουμε περιορισμούς του υλικού από την τροφοδοσία μιας συσκευής, για παράδειγμα ενδέχεται μια συσκευή να πρέπει να τροφοδοτείτε από ηλιακή ενέργεια, άρα δεν μπορεί να έχει πολύ απαιτητικό υλικό.

## 1.3 Μνήμες

Τα προγράμματα και τα δεδομένα προς επεξεργασία πρέπει να αποθηκευτούν στο σύστημα πριν ή κατά τη διάρκεια της επεξεργασίας. Οι μνήμες πρέπει να είναι αρκετά μεγάλες για τις ανάγκες, αρκετά αποδοτικές και να έχουν χαμηλό κόστος, φυσικό μέγεθος και κατανάλωση ενέργειας [3]. Υπάρχει μεγάλο χάσμα στην ταχύτητα του επεξεργαστή και την ταχύτητα της μνήμης παρότι οι ταχύτητες επεξεργαστών μένουν στάσιμες τα τελευταία χρόνια, αυτό σημαίνει ότι υπάρχουν μεγάλες απαιτήσεις όσο αφορά την ταχύτητα μεταφοράς δεδομένων από και προς την μνήμη [3].

### 1.3.1 Ιεραρχία Μνήμης

Οι μνήμες έχουν μια ιεραρχία, όσο πιο ψηλά είναι είναι ένας τύπος μνήμης στην ιεραρχία τόσο πιο γρήγορος είναι. Η ιεραρχία είναι:

- Καταχωρητές Επεξεργαστή
- Κρυφές μνήμες
- Μνήμη RAM (Κύρια Μνήμη)

#### Καταχωρητές - Registers

Οι καταχωρητές είναι πολύ γρήγορη και πολύ μικρή μνήμη που υπάρχει μέσα στον επεξεργαστή. Χρησιμοποιούνται από της εντολές του επεξεργαστή για την εγγραφή και την ανάγνωση δεδομένων κατά την εκτέλεση των προγραμμάτων.

#### Μνήμη Cache

Σήμερα, οι ταχύτητες των επεξεργαστών αυξάνονται με μεγαλύτερο ρυθμό από τις ταχύτητες των μνημών, με αποτέλεσμα οι επεξεργαστές βασίζονται στις κρυφές μνήμες (cache) για να



αυξήσουν την αποδοτικότητα τους [6]. Αυτές οι μνήμες χρησιμοποιούνται ευρέως για την επιτάχυνση της απόδοσης του συστήματος μνήμης [6].

Η αρχιτεκτονική της μνήμης πολύ μεγάλες επιπτώσεις στην απόδοση των εφαρμογών και κατά συνέπεια του συστήματος [3]. Μπορεί να υλοποιηθούν ως ενοποιημένες μνήμες (αρχιτεκτονική Princeton ή von Neumann), δηλαδή οι εντολές και τα δεδομένα αποθηκεύονται στην ίδια μνήμη, και με διαφορετική μνήμη για εντολές και δεδομένα (αρχιτεκτονική Harvard) [5].

Η μνήμες αυτές αποθηκεύουν δεδομένα τα οποία ανήκουν στην κύρια μνήμη και η απόδοση της μετρίεται από το πόσα από τα αιτήματα του επεξεργαστή προς την κύρια μνήμη εξυπηρετούνται από την μνήμη Cache.

Μπορεί να έχουμε πολλά επίπεδα μνήμης σε μια Cache, συνήθως το επίπεδο 1 της μνήμης (Level 1) υλοποιείται εντός του ολοκληρωμένου στον επεξεργαστή ενώ τα υπόλοιπα επίπεδα εκτός του ολοκληρωμένου [6]. Τα επίπεδα μνήμης συνδέονται μεταξύ τους, συνήθως έχουμε 3 και ελέγχεται το κάθε ένα επίπεδο πριν φτάσει το αίτημα για τα δεδομένα που θέλει ο επεξεργαστής στην κύρια μνήμη [5].

### 1.3.2 Μνήμες Τυχαίας Προσπέλασης - Random Access Memory (RAM)

Οι μνήμες τυχαίας προσπέλασης είναι γνωστές και ως κύρια μνήμη, λέγονται μνήμες τυχαίας προσπέλασης γιατί οι διευθύνσεις μπορούν να διαβαστούν με τυχαία σειρά [6].

Οι δύο κύριοι κατηγορίες μνημών τυχαίας προσπέλασης είναι η στατική (SRAM) και η δυναμική (DRAM). Η στατική μνήμη είναι ταχύτερη αλλά καταναλώνει περισσότερη ενέργεια από την δυναμική [6]. Μπορούμε να βάλουμε περισσότερη δυναμική μνήμη σε ένα ολοκληρωμένο αλλά οι τιμές της πρέπει να ανανεώνονται περιοδικά [6].

# Κεφάλαιο 2

## Συστήματα για Κωδικοποίηση Video

Σε αυτό το κεφάλαιο θα αναλυθεί το σύστημα για την αποδοτική κωδικοποίηση εικόνων σε πρότυπο JPEG2000 με παράλληλο Embedded Block Coding and Optimal Truncation (EBCOT) που περιγράφεται στο [7] και το σύστημα που προτείνεται στο Embedded hardware low cost JPEG 2000 video coding system [8] και μετά θα γίνει σύγκριση μεταξύ τους.

### 2.1 Κωδικοποίηση JPEG 2000

Για να αναλυθεί σωστά η αρχιτεκτονική που επεξεργάζεται τα δεδομένα πρέπει πρώτα να αναλυθεί η επεξεργασία των δεδομένων.

Η κωδικοποίηση αυτή χρησιμοποιεί τον διακριτό μετασχηματισμό κυματιδίων (Discrete Wavelet Transform - DWT) για να συμπίεσει εικόνες. Ο αλγόριθμος αυτός μετατρέπει την εικόνα σε μια σειρά σωματιδίων που μπορούν να αποθηκευτούν περισσότερο αποδοτικά [9].

#### 2.1.1 Έννοιες

Υπάρχουν μερικές έννοιες που είναι απαραίτητο να εξηγηθούν για την βέλτιστη κατανόηση της επεξεργασίας και των αρχιτεκτονικών.

**Τα Παράθυρα** είναι μια ομάδα εικονοστοιχείων με ένα κεντρικό εικονοστοιχείο. Ένα παράθυρο μπορεί να έχει διαφορετικά σχήματα όπως ορθογώνιο, τετράγωνο και σταυρό. Το παράθυρο καθορίζει την γειτονιά ενός εικονοστοιχείου.

**Τα Κυματίδια** (Wavelets) είναι πεπερασμένα σήματα που χρησιμοποιούνται για την συμπίεση εικόνων και για την αφαίρεση θορύβου. Συμβολίζονται ως κυματίδια  $x/y$  όπου το  $x$  είναι ο αριθμός των συνιστωσών στο κατωπερατό φίλτρο και  $y$  είναι ο αριθμός των συνιστωσών στο ανωπερατό φίλτρο.

#### 2.1.2 Προ επεξεργασία

Αρχικά όλα τα δείγματα της εικόνας παραμερίζονται στο επίπεδο διακριτού συνημιτόνου και κεντρώνονται στο 0, αν υπάρχουν πολλές συνιστώσες, όπως στα κανάλια RGB, τότε μετατοπίζεται η κάθε μία συνιστώσα [10].

Αν η εικόνα είναι RGB τότε μετατρέπεται σε  $Y'C_bC_r$ , ώστε να βελτιωθεί η αποτελεσματικότητα της συμπίεσης, μετά από αυτή τη διαδικασία τα εικονοστοιχεία της εικόνας διαιρούνται σε ορθογώνιους πίνακες (tiles) που θα επεξεργαστούν ανεξάρτητα ο ένας από τον άλλον [10]. Έπειτα η εικόνα χωρίζεται σε ορθογώνιους πίνακες που θα επεξεργαστούν ξεχωριστά. Σε

περίπτωση που έχουμε πολλές συνιστώσες, δηλαδή η εικόνα είναι έγχρωμη, τότε η κάθε συνιστώσα μπορεί να επεξεργαστεί διαφορετικά [10].

### 2.1.3 Κωδικοποίηση

Μετά τον διαχωρισμό των πινάκων υπολογίζεται ο μονοδιάστατος διακριτός μετασχηματισμός κυματιδίων των γραμμών και των στηλών των συνιστώμενων. Αυτός ο αλγόριθμος στηρίζεται σε μια διορθογώνια κλιμάκωση συντελεστών  $5/3$ , φίλτρο που είναι αναστρεψίμο και σε φίλτρο  $9/7$  που δεν είναι. Στην περίπτωση που υπάρχουν μη ακέραιες τιμές πρέπει να στρογγυλοποιηθούν και να χρησιμοποιηθεί διάνυσμα κυματιδίων κλιμάκωσης συντελεστών  $9/7$ , αυτό το φίλτρο δεν είναι αναστρέψιμο [10].

## 2.2 Κωδικοποίηση με Παράλληλο EBCOT

Σε αυτή την ενότητα περιγράφεται το σύστημα κωδικοποίησης που προτείνεται στο [7]. Ο αλγόριθμος EBCOT χρησιμοποιείται για την κωδικοποίηση JPEG2000, δέχεται κομμάτια της εικόνας μαζί με συνιστώσες κβαντισμού [11].

### 2.2.1 Κωδικοποίηση Επιπέδου Bit

Αρχικά για την κωδικοποίηση σε JPEG χρησιμοποιείται η κωδικοποίηση επιπέδου Bit (Bit Plane Coding), όπου παίρνουμε τις συνιστώσες που προκύπτουν μετά από την μετατροπή κυματιδίων (Wavelet Transform) και τις μετατρέπουμε σε κλίμακα προσημασμένου αριθμού (signed number magnitude) και αποθηκεύονται σε μνήμη τμήματος κώδικα (Code Block Memory) για να επεξεργαστούν ανεξάρτητα. Η κωδικοποίηση ξεκινά από το Most Significant Bit και φτάνει στο Least Significant Bit, κάθε επίπεδο χωρίζεται σε λωρίδες τεσσάρων γραμμών και ελέγχονται ανά στήλη ξεκινώντας από τα αριστερά στα δεξιά και από πάνω προς τα κάτω.

Χρησιμοποιούνται τέσσερις αριθμητικές κωδικοποιήσεις για την κωδικοποίηση των bits:

- Κωδικοποίηση με Μηδενικά (Zero Coding)  
Χρησιμοποιεί την πληροφορία στην γειτονιά του bit για την κωδικοποίηση του.
- Προσημασμένη Κωδικοποίηση (Sign Coding)  
Χρησιμοποιείται όταν ένα bit γίνει το (Most Important Bit) σημαντικό bit στο τωρινό επίπεδο bit (bit plane).
- Magnitude Refinement  
Χρησιμοποιείται για να κωδικοποιηθεί ένα bit που έχει ήδη γίνει το πιο σημαντικό στα προηγούμενα επίπεδα.
- Run Length Coding  
Είναι ένας τρόπος κωδικοποίησης κατά την οποία συνεχόμενα στοιχεία με την ίδια τιμή αποθηκεύονται ως ζευγάρια τιμής και επαναλήψεων [12].

Κάθε δείγμα περνά από 3 φάσεις, το πέρασμα καθαρισμού (Clean Up Pass - CUP), το πέρασμα σημαντικής πέρασμα σημαντικής διάδοσης (Significant Propagation Pass - SPP) και το πέρασμα λέπτυνσης μεγέθους (Magnitude Refinement Pass - MRP).

- Το Significance Propagation Pass  
Κωδικοποιούνται τα bits που δεν είναι στις σημαντικές θέσεις αλλά τουλάχιστον ένα από τους οχτώ γείτονες του είναι.

- To Magnitude Refinement Pass  
Κωδικοποιεί όλα τα bits που έγιναν τα περισσότερο σημαντικά κατά το προηγούμενο πέρασμα.
- To Clean Up Pass  
Είναι το τελευταίο πέρασμα κατά το οποίο κωδικοποιούνται bits που δεν κωδικοποιήθηκαν κατά τις δύο προηγούμενες. Πάντα το αρχικό πέρασμα είναι το Clean Up Pass [11].

### 2.2.2 Συμπίεση

Η συμπίεση γίνεται με τη χρήση ενός MQ Coder που δέχεται το δυαδικό ψηφίο από το Bit Plane Coding και τα συμπραζόμενα που προκύπτουν από τη γειτονιά του. Ως έξοδο επιλέχθηκε η αναπαράσταση των δεδομένων ως LPS (Least Probable Symbol - Λιγότερο Πιθανό Σύμβολο) και MSB (Most Probable Symbol - Περισσότερο Πιθανό Σύμβολο) που αναπαριστούν τις πιθανότητες των συμβόλων. Κάθε δυαδική απόφαση διαιρείται αναδρομικά.

### 2.2.3 Αρχιτεκτονική

Η αρχιτεκτονική του συστήματος πρέπει να είναι τέτοια ώστε να βελτιστοποιεί την διαδικασία της κωδικοποίησης εικόνων με το πρότυπο JPEG2000.

#### Μνήμη

Στην αρχιτεκτονική που έχει σχεδιαστεί η μνήμη χωρίζεται σε έξι μέρη που περιέχουν τις μεταβλητές κατάστασης των λωρίδων, ίδια υποδομή χρησιμοποιείται για τα bit μεγέθους και τις συνιστώσες προσήμανσης (sign coefficients). Σε έναν κύκλο του ρολογιού αποστέλλουν τις 32 στήλες και τους γείτονες τους για να γίνει η επεξεργασία. Μετά την κωδικοποίηση μιας λωρίδας μια ενημερωμένη μεταβλητή κατάστασης διαβάζει από τη μνήμη μεταβλητής κατάστασης σε ένα κύκλο. Αυτή η αρχιτεκτονική επιτρέπει την διεκπεραίωση εγγραφής και ανάγνωσης στον ίδιο κύκλο.

#### Επεξεργαστής

Οι επεξεργαστές που χρησιμοποιεί η αρχιτεκτονική είναι Virtex5 (XC5LX30T) και Virtex4 (XC4VLX80) που είναι FPGA και το Spartan 3A DSP 3400 (XC3SD3400A).

- Virtex5 XC5LX30T  
Είναι ένας επεξεργαστής FPGA με χρονισμό στα 372 MHz. Διαθέτει πίνακα 80x30, 4800 λωρίδες Virtex-5 και μέγιστη μνήμη RAM 320Kb.
- Virtex4 XC4VLX80  
Είναι ένας επεξεργαστής FPGA με χρονισμό στα 328 MHz. Διαθέτει πίνακα 160x56, 35840 λωρίδες Virtex-5 και μέγιστη μνήμη RAM
- Spartan 3A DSP 3400  
Είναι επεξεργαστής FPGA με προσθήκη λωρίδων (slices) από το XtremeDSP™ DSP48A για καλύτερη υποστήριξη εφαρμογών που απαιτούν ψηφιακή επεξεργασία σήματος, όπως το JPEG2000 [13]. Ο επεξεργαστής είναι χρονισμένος στο 189MHz.

## Μνήμη

Το σύστημα έχει δύο διαφορετικές μνήμες, μια για τα τμήματα εικονοστοιχείων που θα επεξεργαστεί το σύστημα και μια για τις συνιστώσες-μεταβλητές που θα χρησιμοποιηθούν για την επεξεργασία.

## Διαδικασία Κωδικοποίησης

Η διαδικασία κωδικοποίησης εκτελεί τις τέσσερις διαφορετικές κωδικοποιήσεις που υπάρχουν ανάλογα με κάποιες συνθήκες.

Οι τέσσερις διεργασίες που πρέπει να γίνουν είναι:

- Το Zero Coding  
Χρησιμοποιεί 9 εικονοστοιχεία και το γενικό περιεχόμενο (context) δημιουργείται από τους γείτονες τους.
- Το Sign Coding  
Είναι διαδικασία δύο βημάτων και χρησιμοποιεί 5 εικονοστοιχεία. Στο πρώτο στάδιο χρησιμοποιούνται οι συνιστώσες των οριζόντιων και κάθετων γειτόνων για να δημιουργηθούν οι οριζόντιες και οι κάθετες συνεισφορές και ένα xor bit. Στο δεύτερο βήμα το γενικό περιεχόμενο σχηματίζεται από τις συνεισφορές και τα δεδομένα από την πράξη xor (exclusive OR) του bit προσήμανσης (sign bit) και του xor bit.
- Το Magnitude Refinement Pass  
Χρησιμοποιεί τρία εικονοστοιχεία για την δημιουργία του γενικού περιεχομένου τα οποία εξαρτώνται από το αν αυτό το πέρασμα χρησιμοποιείται για πρώτη φορά στο εικονοστοιχείο και στους οχτώ άμεσους του γείτονες.
- Το Run Length Coding  
Χρησιμοποιείται μόνο όταν ένα προηγούμενο μη σημαντικό bit έχει γίνει σημαντικό κατά την διάρκεια Zero Coding, Sign Coding ή Run Length Coding. Επίσης η πληροφορία προσήμανσης κωδικοποιείται χρησιμοποιώντας ένα από τα πέντε διαφορετικά γενικά περιεχόμενα που εξαρτώνται στη προσήμανση και την σημαντικότητα των άμεσα οριζόντιων και κάθετων γείτονες.

## Ελεγκτής Μοντέλου

Ο ελεγκτής μοντέλου ελέγχει όλα τα απαιτούμενα σήματα ελέγχου για να επιτρέψει όλα τα επιμέρους συστήματα στην αρχιτεκτονική, δηλαδή ελέγχει το σύστημα όπως ένας μικροελεγκτής. Αυτές οι λειτουργίες περιλαμβάνουν την φόρτωση της λωρίδας από την μνήμη τμημάτων ROM, διάβασμα και γράψιμο στην μνήμη λωρίδων, τον έλεγχο των δεικτών για πρόσβαση στον δημιουργό αλληλουχίας (sequencer) CX/D και την δημιουργία σημάτων για τον κωδικοποιητή Bit Plane.

## Γεννήτρια Πληροφορίας Στηλών

Κατά τη διάρκεια των Zero Coding και Magnitude Refinement Coding ο καταχωρητής που ανταποκρίνεται στο κωδικοποιημένο δείγμα ενδέχεται να ενημερώσει την πληροφορία όταν η λωρίδα κωδικοποιηθεί.

### CX/D Sequencer

Αυτό το τμήμα αποστέλλει τα δεδομένα (CX, D) στον MQ Coder για να κωδικοποιηθούν στη σωστή σειρά. Η επιλογή των συμφραζομένων (Context - CX) γίνεται από το Zero Coding, το Sign Coding, το Magnitude Refinement ή τα Run Length Coding συμφραζόμενα.

### 2.2.4 Απόδοση

Όλα τα δείγματα σε μια λωρίδα επεξεργάζονται σε έναν κύκλο του ρολογιού και χρειάζονται 272 κύκλοι για να κωδικοποιηθεί ένα επίπεδο bit.

Χρησιμοποιημένο FPGA	XC5VLX50T	XC4VLX80	XC3SD3400a
Συχνότητα (MHz)	372	328	189
Είσοδοι LUTs	228	337	340
Χρησιμοποιημένα Slices	201	193	188
Χρησιμοποιημένα FF Slices	26	206	207
Κατανάλωση Ενέργειας (mW)	50 (30.2 °C)	79 (27.3 °C)	12 (30.2 °C)

Table 2.1: Χαρακτηριστικά Επεξεργαστών

Από τον πίνακα 2.1 φαίνονται οι συχνότητες, οι εόσοδοι LUT, που το μικρότερο είναι καλύτερο, τα χρησιμοποιημένα κομμάτια (slices) που είναι τα κομμάτια του επεξεργαστή που χρησιμοποιήθηκαν για την επεξεργασία, τα κομμάτια FF (Flip Flop) που χρησιμοποιήθηκαν και η κατανάλωση ενέργειας, όπου όσο χαμηλότερη τόσο το καλύτερο.

Αρχιτεκτονική	[14]	[15]	Προτεινόμενο
Συχνότητα (MHz)	50	67	186
Είσοδοι LUTs	7071	2488	340
Χρησιμοποιημένα Slices	4420	2149	186
Χρησιμοποιημένα FF Slices	1560	105	27
Χρησιμοποιημένο FPGA	XC2V1000-6		

Table 2.2: Σύγκριση Με Υπάρχοντα Συστήματα

Στον πίνακα 2.2 φαίνεται πως το προτεινόμενο σύστημα παρέχει καλύτερη απόδοση από άλλα συστήματα EBCOT.

Εικόνα	LZBP	Επεξεργασμένα Bit Planes	Κύκλοι ανά Εικόνα	Ρυθμός Επεξεργασίας
Lena	918	1390	366960	1.514
Baboon	587	1717	453288	2.925
Peppers	758	1546	408144	2.039
Barbara	788	1361	459304	1.727

Table 2.3: Απόδοση κύκλων για κωδικοποίηση με τμήμα κώδικα 32 επί 32

Στον πίνακα 2.3 φαίνεται η απόδοση του συστήματος όσο αφορά την κωδικοποίηση ορισμένων εικόνων, φαίνεται ότι δεν είναι κατάλληλο για κωδικοποίηση εικόνων για βίντεο.

## 2.3 Σύστημα για Κωδικοποίηση Βίντεο με Χρήση JPEG2000

Σε αυτό το κεφάλαιο θα μελετηθεί ένα σύστημα για την αποδοτική κωδικοποίηση βίντεο χρησιμοποιώντας τον αλγόριθμο για κωδικοποίηση εικόνων JPEG 2000 σε ενσωματωμένα συστήματα που προτείνεται στο [8].

### 2.3.1 Υλοποίηση του Μετασχηματισμού

Η υλοποίηση του μετασχηματισμού γίνεται χρησιμοποιώντας έναν επεξεργαστή για ψηφιακά σήματα ADSP-BF533 Blackfin (Digital Signal Processor - DSP) μαζί με την κάρτα εξομίωσης EZ-KIT LITE και δύο ολοκληρωμένα για την συμπίεση JPEG2000 ADV202. Επίσης χρησιμοποιείται το ADV7183 αποκωδικοποιητής. Αυτό το σύστημα χρησιμοποιεί τον χωρικό πλεονασμό για να βελτιώσει την απόδοση της κωδικοποίησης.

#### ADV202

Το ολοκληρωμένο ADV202 χρησιμοποιείται για την αποδοτική κωδικοποίηση του JPEG2000. Περιέχει έναν επεξεργαστή RISC, μια μηχανή για την μετατροπή κυματιδίων, τρεις κωδικοποιητές εντροπίας και ένα σύστημα άμεσης πρόσβασης μνήμης (Direct Memory Access - DMA) [16]. Δέχεται ένα ορθογώνιο κομμάτι από εικονοστοιχεία και αμέσως μετά εκτελεί την μετατροπή κυματιδίων και αποθηκεύει τις συνιστώσες στην μνήμη. Μετά εκτελεί την κωδικοποίηση εντροπίας όπου κωδικοποιεί την εικόνα για να υπακούει στο πρότυπο JPEG2000. Με κωδικοποίηση 5/3 μπορεί να πετύχει ρυθμό 40 εκατομμυρίων δειγμάτων το δευτερόλεπτο ενώ με 9/7 μπορεί μεγαλύτερο ρυθμό.

Όσο αφορά τις διεπαφές έχει FIFO (First in First Out) Buffers για να αποστέλλει δεδομένα μέσω ενός υψηλής ταχύτητας διαύλου.

**Ο επεξεργαστής** που περιέχει χρησιμοποιείται για την διαχείριση των επιμέρους κομματιών καθώς και το parsing και την δημιουργία των δεδομένων JPEG2000. Περιέχει μνήμη για εντολές και δεδομένα, ελεγκτή διακοπών, διεπαφές για διαύλους και άλλα συστήματα όπως χρονιστές και μετρητές.

**Η μνήμη** χρησιμοποιείται κυρίως για την αποθήκευση των συνιστώμενων των κυματιδίων, τη ροή κώδικα JPEG2000. Επίσης μπορεί να χρησιμοποιηθεί και για την αποθήκευση εντολών και δεδομένων του επεξεργαστή όταν χρειαστεί.

**Το σύστημα μνήμης** προσφέρει μεγάλο εύρος ζώνης για επικοινωνία στα επιμέρους τμήματα του ολοκληρωμένου και όχι μόνο στη μνήμη. Είναι πολύ σημαντικό κομμάτι του συστήματος καθώς η ταχύτητα επικοινωνίας πρέπει να είναι αρκετή ώστε να μην υπάρχει bottleneck.

**Η κωδικοποίηση** μπορεί να γίνει με δύο διαφορετικούς τρόπους, τα εικονοστοιχεία μπορούν να περάσουν μέσα από τον HDATA δίαυλο, που μετά συνδέεται στο HIPI (Host Interface - Pixel Interface) που κάνει την συμπίεση των δεδομένων. Διαφορετικά μπορεί να χρησιμοποιηθεί ο δίαυλος VDATA που χρησιμοποιεί τα δεδομένα από εικονοστοιχεία ή βίντεο. Στις δύο περιπτώσεις τα δεδομένα περνάνε στο HIPI. Τα δεδομένα από τον επεξεργαστή περνάνε στο ADV202, μετά στη διεπαφή εικονοστοιχείων και μετά γίνεται η μεταμόρφωση κυματιδίων από γραμμή σε γραμμή, άρα τα δεδομένα που μπαίνουν πρέπει να αφαιρεθούν τα επίπεδα (de-interleave). Μετά από αυτό τα δεδομένα αποσυντέθονται σε υποζώνες και μετά πραγματοποιείται η μετατροπή κυματιδίων, αφότου ολοκληρωθεί η μετατροπή τα δεδομένα αποθηκεύονται στην εσωτερική μνήμη και μετά τα δεδομένα μεταφάιρονται στη μονάδα συμπίεσης. Μετά περνάνε από τρία στάδια για την κωδικοποίηση:

- Κβαντοποίηση (Quantization)  
Γίνεται η κβαντοποίηση για να μειωθεί το πλήθος των δεδομένων
- Βελτιστοποίηση Διαστρέυσης Ρυθμού (Rate Distortion Optimization)  
Το ποσό της κβαντοποίησης Βασίζεται στις απαιτήσεις ρυθμού ή τις απαιτήσεις ποιότητας

- Μοντελοποίηση Συμφραζομένων (Context Modeling)  
Αποδίδει πληροφορία σχετικά με την σημαντικότητα (significance) μιας μεμονομένης συνιστώσας.

Μετά από αυτό τα δεδομένα είναι πλήρως συμβατά με το πρότυπο JPEG 2000.

Έπειτα η περεταίρω κωδικοποίηση γίνεται από το εξωτερικό σύστημα και την διεπαφή ξενιστή (host) που χρησιμοποιείται με βάση ένα πρωτόκολλο ανάγνωσης εγγραφής. Όσο αφορά την αποκωδικοποίηση τα δεδομένα αποθηκεύονται στην εσωτερική μνήμη και μετά περνώνται στον κωδικοποιητή εντροπίας που αποκωδικοποιούν τη ροή δεδομένων.

### Επεξεργαστής

Ο επεξεργαστής που έχει το σύστημα είναι ο Blackfin ADSP-BF533. Περιέχει την είσοδο του βίντεο, επεξεργασία των δεδομένων του βίντεο, αριθμητικές πράξεις, πρόσβαση στη μνήμη και το σύστημα του διαύλου. Αυτός ο επεξεργαστής έχει λειτουργία και ως μικροελεγτή, που σημαίνει ότι μπορεί να κάνει επεξεργασία σήματος και διαχείριση του συστήματος ταυτόχρονα. Το σύνολο εντολών που χρησιμοποιεί είναι RISC 32 δυαδικών ψηφίων (bit) και ένα σύνολο εντολών 8 διαδικών ψηφίων ειδικά για βίντεο.

Η διεπαφή βίντεο του επεξεργαστή είναι μια αμφίδρομη παράλληλη περιφερειακή διεπαφή μπορεί να μεταφέρει 8 ή 16 δυαδικά ψηφία, λαμβάνει με ταχύτητα μέχρι 65MHz και αποστέλλει από τον επεξεργαστή με ταχύτητα μέχρι και 60MHz. Το γεγονός ότι είναι αμφίδρομη οφείλεται στο ότι ο επεξεργαστής έχει μόνο μια τέτοια διεπαφή και κατά συνέπεια ενδέχεται να υπάρξουν συγκρούσεις δεδομένων και καθυστέρηση.

Σε αυτό το σύστημα χρησιμοποιείται για την μεταφορά πλαισίων στο ADV202, τον υπολογισμό των διαφορικών πλαισίων και τη στατιστική ανάλυση στα διαφορικά πλαίσια. Το μειονέκτημα με το γεγονός ότι ο επεξεργαστής δημιουργεί τα διαφορικά πλαίσια είναι ότι δεν μπορούν τα δεδομένα να πάνε απ' ευθείας στο ADV202 και πρέπει να αποθηκευτούν πρώτα.

**Ο ελεγκτής για άμεση πρόσβαση μνήμης** (Direct Memory Access) επιτρέπει την εσωτερική μεταφορά δεδομένων στη μνήμη και την μεταφορά μεταξύ μνήμης και περιφερειακού χωρίς τη διαμεσολάβηση επεξεργαστή, επίσης ο πυρήνας επιτρέπει ταυτόχρονες προσπελάσεις.

Δεν γίνεται εγγραφή δεδομένων σε τμήματα μνήμης. Η μνήμη χωρίζεται σε δύο κομμάτια, την εσωτερική και την εξωτερική. Στην εσωτερική μνήμη ακολουθείτε ιεραρχικό μοντέλο:

1. L1 Static Random Access Memory (SRAM)
2. Memory Mapped Registers
3. Boot Read Only Memory

Η μνήμη L1 έχει την αρχιτεκτονική Harvard που σημαίνει ότι έχει διαφορετική μνήμη για τις εντολές και για τα δεδομένα, αυτό επιτρέπει 4 προσπελάσεις ανά κύκλο του ρολογιού [17]. Έχει μέγεθος 16K bytes και ένα κομμάτι της μνήμης μπορεί να χρησιμοποιηθεί ως μνήμη cache, επίσης υπάρχει η μνήμη SRAM που είναι για εντολές η οποία έχει μέγεθος 64K bytes [17]. Πέρα από τις μνήμες για εντολές έχουμε και τις μνήμες για δεδομένα, που έχουμε μια μνήμη SRAM μεγέθους 32K Byte που μπορεί να χρησιμοποιηθεί για Cache, μια μνήμη SRAM για δεδομένα και μια μνήμη scratchpad SRAM με μέγεθος 2K, αυτή η μνήμη χρησιμοποιείται για την αποθήκευση προσωρινών δεδομένων όπως δεδομένα από υπολογισμούς [17].

Αυτός ο επεξεργαστής υποστηρίζει μια μνήμη SDRAM που μπορεί να έχει μέγεθος από 16MB μέχρι και 128MB. Μετά περιέχει τέσσερα κανάλια που έχουν μνήμη 1MB και αφορούν ασύγχρονες μνήμες.



Ο μηχανισμός διαχείρισης μνήμης στον επεξεργαστή βασίζεται στις σελίδες, επίσης προσφέρει έλεγχο σχετικά με τη δυνατότητα caching ενός εύρους μνήμης καθώς και την διαχείριση ιδιοτήτων προστασίας σε επίπεδο σελίδας.

Η αρχιτεκτονική προσφέρει τέσσερα διαφορετικά μήκη για μια σελίδα:

- 1KByte
- 4KByte
- 1MByte
- 4MByte

### Κωδικοποίηση

Αρχικά ο επεξεργαστής παίρνει και επεξεργάζεται τα δεδομένα βίντεο και μετά τα αποθηκεύει στο ολοκληρωμένο ADV202.

Για την κωδικοποίηση βίντεο το αρχικό πλαίσιο διαβάζεται και χρησιμοποιείται ως προθεματικό (πλαίσιο αναφοράς) και μετά χρησιμοποιούνται διαφορεικά πλαίσια μεταξύ του τωρινού που θα κωδικοποιηθεί και του ανακατασκευασμένου πλαισίου αναφοράς, αυτό γίνεται σε προσαρμοστικά σύνολο εικόνων στο βίντεο, δηλαδή για ένα σύνολο θα έχουμε διαφορετικό πλαίσιο αναφοράς, το πλήθος των εικόνων σε ένα σύνολο εικόνων αποφασίζεται από στατιστικές μεθόδους. Τα διαφορετικά πλαίσια βασίζονται στην αριθμητική διαφορά δύο εικονοστοιχείων στην ίδια θέση. Στην περίπτωση βίντεο με υψηλή κίνηση δουλεύει καλύτερα ένας μικρός αριθμός εικόνων ανά σύνολο εικόνων, όμως αυτό απαιτεί πολλά πλαίσια αναφοράς που κάνει την κωδικοποίηση δύσκολη.

Για την εκτίμηση της απόδοσης χρησιμοποιείται το Peak Signal to Noise Ratio (PSNR) σε συνιστώσα με την ομαλότητα της εικόνας που έχουν αρνητική συσχέτιση με -0.95. Όσο μεγαλύτερο το PSNR έχουμε τόσο μικρότερο θόρυβο και τόσο καλύτερο αποτέλεσμα.

PSNR	MJPEG200	cGOP	oGOP	Dynamic Coder
25	32.72	78.44	78.44	93.94
27	23.85	54.62	54.62	86.85
29	18.23	37.69	38.41	40.03
32	14.56	22.69	26.49	24.87
37	7.79	15.36	15.36	16.74
39	6.21	11.05	11.05	12.32

Table 2.4: Ταχύτητα Συμπίεσης για Στόχο PSNR

### 2.3.2 Υλοποίηση Υλικού

Στην εργασία παρουσιάζονται τρεις διαφορετικές υλοποιήσεις του υλικού αυτού.

#### Βασική Υλοποίηση

Αρχικά ορίζουμε τέσσερις buffers για εικόνες, οι τρεις πρώτες αποθηκεύουν δεδομένα ενώ η τέταρτη αποθηκεύει τα ανακατασκευασμένα πλαίσια ως πλαίσια αναφοράς. Τα εισερχόμενα δεδομένα θα μεταφέρονται στη μνήμη του επεξεργαστή Blackfin ενώ τα διαφορετικά πλαίσια υπολογίζονται και μεταφέρονται στο ολοκληρωμένο ADV202 για συμπίεση. Η διαδικασία της καταχώρησης μνήμης εκπληρώνεται από τον μηχανισμό για άμεση πρόσβαση μνήμης.

### Δεύτερη Υλοποίηση

Σε αυτή την υλοποίηση τοποθετήθηκαν δύο συστήματα άμεσης πρόσβασης μνήμης (DMAs) ώστε να επιταχυνθούν οι μεταφορές της μνήμης (MEMDMA). Το MEMDMA χρειάζεται την διεύθυνση λήψης, την διεύθυνση αποστολής και τον αριθμό των bits που πρέπει να μεταφερθούν. Οι καταχωρητές DMA προσπελάσσονται μέσα από μια διεύθυνση που περιέχει τις κύριες ρυθμίσεις του περιφερειακού όπως έγινε με το HIPI. Για την κωδικοποίηση βίντεο τα δεδομένα γράφονται στην διεπαφή εικονοστοιχείων και τα συμπιεσμένα δεδομένα αποστέλλονται μέσω της Code FIFO. Μετά από την εγκατάσταση του HIPI τα αρχικά δεδομένα αποστέλλονται στη διεπαφή εικονοστοιχείων. Μετά τα δεδομένα δρομολογούνται μέσα από την γραμμή επικοινωνίας του JPEG 2000 και μετά τα συμπιεσμένα συμβατά με JPEG 2000 δεδομένα μπορούν να γραφούν στο CODE FIFO στο buffer βίντεο μέχρι να μην υπάρχουν άλλα.

### Τρίτη Υλοποίηση: Με Χρήση Κρυφής Μνήμης

Η κρυφή μνήμη (Cache) είναι διαθέσιμη κάτω από τον έλεγχο ενός ελεγκτή στον επεξεργαστή Blackfin, επίσης η πρόσβαση μπορεί να γίνει σε έναν κύκλο του ρολογιού, δηλαδή πολύ γρήγορα. Η εξωτερική μνήμη χωρίζεται σε τμήματα και η μνήμη cache χωρίζεται περεταίρω σε υποτμήματα. Ο επεξεργαστής προσπελαύνει την κρυφή μνήμη (cache line fill) όταν χρειάζεται να φορτώσει μια εγγραφή από την εξωτερική μνήμη και μετά η γραμμή προωθείται στον επεξεργαστή, συχνά χρειάζεται να φορτώσει 32 ψηφία (bits) από την μνήμη. Η υλοποίηση αυτή επιτρέπει στον CPU να επεξεργαστεί την γραμμή από την cache όσο αυτή φορτώνεται αντί να περιμένει την ολική φόρτωση της στη μνήμη.

Αν έχουμε cache miss, δηλαδή η μνήμη δεν διαθέτει τα δεδομένα που θέλει ο επεξεργαστής τότε η μονάδα εντολών μνήμης (instruction memory unit) κάνει μια πρόσβαση στη μνήμη για να πάρει το εκλειπόμενο τμήμα, όσο γίνεται αυτό ο επεξεργαστής περιμένει. Όταν πρόκειται η μνήμη να διαγράψει μια γραμμή χρησιμοποιεί τον αλγόριθμο λιγότερο πρόσφατης χρήσης (Least Recently Used - LRU) για να αποφασίσει πια γραμμή θα διαγράψει.

Κατά τη διάρκεια της φόρτωσης μιας γραμμής από τη μνήμη κάθε 64 ψηφία κρατώνται σε έναν buffer γραμμών και μετά γράφονται σε ένα κανάλι μνήμης 4KB. Το buffer επιτρέπει στον επεξεργαστή να επεξεργαστεί τα δεδομένα όπως αποθηκεύονται.

Ο επεξεργαστής BF533 έχει 64KB ενσωματωμένης μνήμης εκ' των οποίων τα 32 είναι διαθέσιμα ως δύο ανεξάρτητα κανάλια μνήμης των 16KB που μπορούν να ρυθμιστούν ως μνήμη SRAM δεδομένων ή εντολών (data or instruction). Μετά τα τμήματα μνήμης 16KB χωρίζονται σε τέσσερα υπο κανάλια των 4KB όπου κάθε υποκανάλι χωρίζεται σε 64 τμήματα όπου κάθε τμήμα της μνήμης χωράει τέσσερις γραμμές 32 δυαδικών ψηφίων. Το τμήμα του επεξεργαστή που είναι υπεύθυνο για την διαχείριση της μνήμης cache είναι ο descriptor cacheability protection lookaside buffer (CPLB). Όταν είναι ενεργοποιημένος ο descriptor τότε όλες οι θέσεις μνήμης που προσπελάσσονται πρέπει να έχουν έναν σχετικό ορισμό σελίδας διαθέσιμο. Το τμήμα προστασίας μνήμης χρησιμοποιείται για προστασία πρόσβασης σε διεργασίες γραφής και ανάγνωσης καθώς και για τους ορισμούς cache και μη cache μνήμης.

### 2.3.3 Απόδοση Συστημάτων

Σε αυτή την υποενότητα θα συγκριθούν των υλοποιήσεων του συστήματος, επίσης συγκρίνεται με μια υλοποίηση Motion JPEG2000 (MJ2K). Η σύγκριση γίνεται σε 34 frames. Ο έλεγχος γίνεται με ένα φορητό media player με ανάλυση 720 επί 578 εικονοστοιχεία και ένα αισθητήρα βίντεο.

### Βασική Υλοποίηση

Στον πίνακα 2.5 φαίνεται η απόδοση του συστήματος όσο αφορά τον χρόνο υπολογισμού κάθε τμήματος του συστήματος σε ms.

	Test 1	Test 2	MJ2K Test 1	MJ2K Test 2
Non-File I/O				
BF533	57.71	56.99	67.74	65.24
ADV202	34.96	35.6	32.25	34.76
Diff. Frame	2.89	2.91	0	0
Indicator 1	2.37	2.39	0	0
Indicator 2	2.07	2.11	0	0

Table 2.5: Απόδοση του Βασικού Συστήματος

Πιο χρήσιμο μετρικό είναι οι εικόνες που μπορούν να κωδικοποιηθούν το δευτερόλεπτο που φαίνεται στον πίνακα 2.6. Φαίνεται η υλοποίηση MJ2K είναι πολύ πιο αποδοτική από την υλοποίηση του συστήματος, ο κύριος λόγος γι' αυτή την καθυστέρηση είναι οι χρόνοι μεταφοράς μεταξύ του ADV202 και του BF533 καθώς και τα ανακατασκευασμένα πλαίσια για επόμενους υπολογισμούς. Για την επίλυση του προβλήματος πρέπει να βελτιωθεί η απόδοση της μνήμης, δηλαδή πρέπει να ανασχεδιαστεί η μνήμη καθώς και ο αγωγός κωδικοποίησης (encoding pipeline). Κάτι τέτοιο μπορεί να γίνει με την μείωση των Framebuffers, τη αύξηση των δεδομένων στο caching καθώς και την χρήση πολλών DMAs (MEMDMA).

	Test 1	Test 2	MJ2K Test 1	MJ2K Test 2
	time/frame	time/frame	time/frame	time/frame
Debug	2700	2500	830	889
Release	428	484	102	102

Table 2.6: Απόδοση του Βασικού Συστήματος

### Δεύτερη Έκδοση: DMAs και MEMDMAs

Η βελτίωση αυτού του συστήματος είναι η χρήση αμφίδρομης επικοινωνίας μνήμης προς μνήμη (MEMDMA) που αποφεύγει τις απ' ευθείας προσπελάσεις εκτός του εύρους του SDRAM. Τα διαφορετικά πλαίσια του BF533 δεν μπορούν να μεταφερθούν απ' ευθείας στο ADV202, πρέπει πρώτα να αποθηκευτούν στη μνήμη.

Στον πίνακα 2.7 φαίνεται η απόδοση του συστήματος MJ2K και του κωδικοποιητή με DMA και χωρίς. Το DMA κάνει μεγάλη διαφορά στην υλοποίηση του MJ2K, καθώς από 6 fps χωρίς DMA πάει στα 20 fps.

	MJ2KDMA	MJ2K	CoderDMA	Coder
Test 1				
Fps (in ms)	20	6	3	2
Time/frame	48	180	320	540
Test 2				
Fps (in ms)	19	6	3	2
Time/frame	63	200	320	510

Table 2.7: Απόδοση του Βασικού Συστήματος σε Σύγκριση με το MJ2K

DMA	Cache	Cycles	%	Time(ms)
-	-	315.385	100	368
+	-	239.687	76	257
+	+	151.069	47.9	142

Table 2.8: Απόδοση των Προτεινόμενων Συστημάτων

### Τρίτη Έκδοση: Cache

Η cache προστέθηκε στο σύστημα για να βελτιωθεί η ταχύτητα της μνήμης. Προστέθηκε cache στο αρχικό σύστημα καθώς και στο σύστημα με MDA. Η απόδοση του φαίνεται στον πίνακα 2.8. Τελικά το σύστημα δεν είναι αρκετό για αποδοτική κωδικοποίηση πραγματικού χρόνου αλλά πιθανόν είναι καλό για άλλες εφαρμογές.

## 2.4 Σύγκριση Συστημάτων

Σε αυτή την ενότητα θα συγκριθούν τα συστήματα που παρουσιάστηκαν στις δύο προηγούμενες ενότητες. Και τα δύο αφορούν την αποδοτική κωδικοποίηση εικόνων χρησιμοποιώντας τον αλγόριθμο JPEG2000.

### 2.4.1 Αρχιτεκτονική

Αρχικά η υλοποίηση του EBCOT χρησιμοποιεί επεξεργαστές FPGA για την υλοποίηση της κωδικοποίησης ενώ το δεύτερο σύστημα χρησιμοποιεί έναν επεξεργαστή DSP καθώς και δύο ολοκληρωμένα αποκλειστικά για την κωδικοποίηση του JPEG2000. Φαίνεται πως το σύστημα με επεξεργαστή και τα ADV202 είναι το πιο ακριβό και το πιο πολύπλοκο.

Το σύστημα με παράλληλο EBCOT χρησιμοποιεί πολύ ειδικευμένο υλικό για ορισμένες από τις λειτουργίες, πιο συγκεκριμένα την κωδικοποιήσεις, την εξαγωγή των γειτονιών εικονοστοιχείων καθώς και τον μηχανισμό για την δημιουργία της ακολουθίας συμφραζομένων και απόφασης (CX/D Sequencer). Όλα αυτά ελέγχονται από έναν ελεγκτή μοντελοποίησης συμφραζομένων. Η αρχιτεκτονική του δεύτερου συστήματος βασίζεται σε έναν επεξεργαστή DSP και δύο ολοκληρωμένα ADV202 για την κωδικοποίηση JPEG2000.

Η αρχιτεκτονική με EBCOT είναι πιο εξειδικευμένη πάνω στο πρόβλημα καθώς είναι υλοποιημένο σε FPGA, δηλαδή σχεδιασμένο αποκλειστικά για το JPEG 2000, όμως το δεύτερο σύστημα είναι και αυτό πολύ εξειδικευμένο πάνω στο JPEG2000 λόγω των ADV202.

### 2.4.2 Απόδοση

Όσο αφορά την απόδοση φαίνεται από τις αποδόσεις η υλοποίηση του δεύτερου συστήματος με DMA και MEMDMA είναι η βέλτιστη. Στον πίνακα 2.9 φαίνεται ο αριθμός των κύκλων

Σύστημα	Κύκλοι ανά Επεξεργασία
Παράλληλο EBCOT	366960
Απλή Υλοποίηση	315385
Υλοποίηση με MEMDMA	239687
Υλοποίηση με Cache	151069

Table 2.9: Απόδοση των Προτεινόμενων Συστημάτων

για να κωδικοποιηθεί μια εικόνα. Η εικόνα ή τα χαρακτηριστικά της, όπως η εντροπία ή

η ανάλυση, δεν αναφέρεται άρα η σύγκριση μπορεί να είναι λίγο άδικη. Φαίνεται όμως αρκετά μεγάλη διαφορά μεταξύ των υλοποιήσεων με παράλληλο EBCOT και την υλοποίηση του δεύτερου συστήματος με Cache, άρα ανεξάρτητα από την εικόνα που χρησιμοποιήθηκε η υλοποίηση με Cache είναι η πιο καλή.

## Κεφάλαιο 3

# Κωδικοποίηση Βίντεο για το Youtube

Πρόσφατα η Google ανακοίνωσε την δημιουργία ειδικών ολοκληρωμένων για την κωδικοποίηση των βίντεο που εξυπηρετεί, αυτό το υλικό λέγεται Video (trans)Coding Unit (VCU) [18], αυτά τα ολοκληρωμένα δεν προορίζονται για τη χρήση σε μικρούς υπολογιστές αλλά σε κέντρα δεδομένων για την επεξεργασία πολύ μεγάλου όγκου δεδομένων ώστε να εξυπηρετηθούν οι πελάτες [19]. Οι απαιτήσεις του youtube για την κωδικοποίηση είναι τεράστιες καθώς περισσότερες από 500 ώρες βίντεο αποθηκεύονται κάθε λεπτό [18]. Πέρα από αυτά πρέπει να εξυπηρετηθούν ανάγκες για απ' ευθείας μετάδοση (Youtube Live) και οι εικόνες και τα βίντεο που ανεβαίνουν σε Google Drive και Google Photos, υπηρεσίες που έχουν πολύ διαφορετικές απαιτήσεις [19]. Όπως φαίνεται αυτό το σύστημα είναι πολύ πιο πολύπλοκο από τα προηγούμενα και κατά συνέπεια αξίζει να γίνει αναφορά σε ξεχωριστό κεφάλαιο.

Αυτό το σύστημα πρέπει να μπορεί να κωδικοποιήσει πολλές και διαφορετικές κωδικοποιήσεις όπως VP9 και AV1 καθώς υπάρχουν πολλές διαφορετικές συσκευές που θα χρησιμοποιούν το youtube η καθεμία με διαφορετικές υπολογιστικές ικανότητες, σε συνδυασμό με τις πολλές αναλύσεις σημαίνει ότι θα υπάρχουν τεράστιες υπολογιστικές απαιτήσεις [19].

### 3.1 Σύστημα Επεξεργασίας

Η πιο σημαντική δουλειά που θα κάνει το σύστημα είναι το Transcoding των βίντεο που περιλαμβάνει προβλήματα όπως την ανάγκη για διαφορετικές αναλύσεις και κωδικοποιήσεις. Επίσης πρέπει τα συστήματα να είναι κλιμακώσιμα, να είναι παράλληλα και η απόδοση να είναι καλή καθώς και το κόστος να είναι χαμηλό.

Για την κάλυψη αυτών των αναγκών το σύστημα έχει σχεδιαστεί με ένα τμήμα επιτάχυνσης υλικού (hardware accelerator building block) και μια μονάδα κωδικοποίησης βίντεο (Video Coding Unit - VCU) σχεδιασμένα μαζί ώστε να βελτιστοποιηθεί η λειτουργία τους σε κατανεμημένες συστάδες με προγραμματιστές (schedulers) μεγάλης κλίμακας (warehouse scale). Ένα κεντρικό κομμάτι του σχεδιασμού είναι η συνσχεδίαση υλικού και λογισμικού.

Οι αρχές σχεδιασμού που ακολουθήθηκαν είναι:

- **Μεγιστοποίηση Χρήσης**  
Τα συστήματα είναι σχεδιασμένα για να αποφευχθεί η υποχρησιμοποίηση τους, δηλαδή όλοι οι πυρήνες να έχουν τον μέγιστο δυνατό φόρτο.
- **Βελτιστοποίηση για την Ανάπτυξη σε Κλίμακα**  
Οι επιταχυντές (accelerators) είναι σχεδιασμένοι για την εκτέλεση σε user space, καθώς η υλοποίηση σε kernel space θα σημαίνει μη διαθεσιμότητα της μηχανής.
- **Σχεδιασμός για Ευελιξία και Προσαρμοστικότητα**  
Πρέπει τα συστήματα να μπορούν να υποστηρίξουν εφαρμογές και απαιτήσεις που μπορεί

να προκύψουν σε μεταγενέστερα στάδια στη ζωή του συστήματος. Άρα έχει σχεδιαστεί η δυνατότητα προγραμματισμού και δια λειτουργικότητας στο υλικό.

### 3.1.1 Ολικός Σχεδιασμός Συστήματος

Κάθε συστάδα μηχανών λειτουργεί ξεχωριστά και έχει μηχανές με VCU και μηχανές χωρίς επιταχυντή και κάθε μηχανή με VCU έχει πολλούς επιταχυντές και ο κάθε επιταχυντής έχει πολλές μηχανές VCU και η κάθε VCU περιέχει πολλά VCU ASICs, αυτό γίνεται ώστε να είναι πιο δύσκολο να περιοριστεί η διεκπεραιωτικότητα λόγω έλλειψης πόρων της μηχανής. Τα VCU ASICs έχουν πολλούς κωδικοποιητές και αρκετούς αποκωδικοποιητές, επίσης περιέχουν Network on Chip και εύρος ζώνης DRAM ώστε να βελτιωθεί η διεκπεραιωτικότητα και η χρήση του. Επίσης το κάθε μηχανήμα που έχει VCU δεν τρέχει άλλες διεργασίες ώστε να προκαλεί προβλήματα με την διαθεσιμότητα των πόρων.

#### Σχεδιασμός των ASICs

Γενικά στην λειτουργία του συστήματος το πιο ακριβό κομμάτι είναι η κωδικοποίηση, κατά συνέπεια το μεγαλύτερο βάρος είναι στην κωδικοποίηση. Μετά από την κωδικοποίηση σε υπολογιστικό κόστος είναι η αποκωδικοποίηση άρα είναι λογικό να βελτιστοποιηθεί για αποκωδικοποίηση μετά. Επιπλέον τα ολοκληρωμένα έχουν σχεδιαστεί για βελτίωση απόδοσης και βελτιώσεις ποιότητας μετά από την ανάπτυξη του συστήματος.

### 3.1.2 Κεντρικός Σχεδιασμός Κωδικοποιητή VCU

Ο κωδικοποιητής VCU είναι το κύριο στοιχείο των ASICs και μπορεί να κωδικοποιήσει H.264 και VP9. Μερικές από τις ιδιότητες είναι η αγωγή (pipelining) εντολών, τοπική αποθήκευση αναφορών για την εκτίμηση κίνησης και άλλες καταστάσεις, επιτάχυνση κωδικοποίησης εντροπίας αλλά βελτιστοποιημένη για κέντρα δεδομένων, ανάπτυξη και στόχους κατανάλωσης επίδοσης και χώρου.

Το βασικό στοιχείο για τον υπολογισμό με αγωγούς είναι είτε ένα 16x16 μακροτμήμα (macroblock) για H.264 ή ένα 64x64 υπερτμήμα VP9, που είναι το μεγαλύτερο τμήμα εικονοστοιχείων στο οποίο λειτουργεί ο κωδικοποιητής.

#### Στάδια Αγώγισης Εντολών

Η αγωγή εντολών (pipelining) έχει τρία βασικά στάδια.

**Στο πρώτο στάδιο** υλοποιεί τα αρχικά στάδια των αλγορίθμων κωδικοποίησης που χρησιμοποιούν τμήματα όπως την εκτίμηση της κίνησης, τμηματοποίηση υπό τμημάτων και τη βασισμένη σε διαστρέυλωση του ρυθμού (rate distortion) μετατροπή και πρόβλεψη, αυτό το στάδιο είναι μακράν το πιο απαιτητικό από άποψη εύρους ζώνης μνήμης.

**Το δεύτερο στάδιο** περιλαμβάνει την κωδικοποίηση εντροπίας για το τμήμα εξόδου, την αποκωδικοποίηση του μακροτμήματος και τοπικό φιλτράρισμα για την δημιουργία των εναλλακτικών πλαισίων για το VP9. Αυτό το στάδιο βασίζεται πολύ σε ακολουθιακή λογική, που σημαίνει ότι η έξοδος του κυκλώματος εξαρτάτε από μια σειρά των προηγούμενων εισόδων και κατά συνέπεια είναι πολύ δύσκολο να υλοποιηθεί σε υλικό. Το χωρικό φιλτράρισμα που χρησιμοποιεί την εκτίμηση της κίνησης από 3 πλαίσια 16 επί 16 για να δημιουργήσει καινούργια φιλτραρισμένα τμήματα με μικρό χωρικό θόρυβο. Αυτή η εκτίμηση μπορεί να χρησιμοποιηθεί για

να δημιουργηθούν εναλλακτικά πλαίσια αναφοράς, αυτά είναι πλαίσια που αποκωδικοποιούνται αλλά δεν εμφανίζονται ποτέ στον χρήστη και χρησιμοποιούνται εξ' ολοκλήρου ως αναφορά για άλλα πλαίσια. Ένα εναλλακτικό πλαίσιο μπορεί να δημιουργηθεί για περισσότερα από 3 πλαίσια, αλλά αυτό έχει επιπτώσεις στην ποιότητα.

**Το τρίτο στάδιο** αρχικά παίρνει την αποκωδικοποιημένη έξοδο του τμήματος κωδικοποίησης που απαιτεί πρόσβαση στα διαγώνια και ανώτερα τμήματα που είναι αποθηκευμένα στην τοπική SRAM. Μετά εφαρμόζει κυκλικό φιλτράρισμα και μη απολεστική συμπίεση buffer με έναν ιδιόκτητο (proprietary) αλγόριθμο που ελαχιστοποιεί το εύρος ζώνης μνήμης ενώ είναι αρκετά ταχύς ώστε να μην αποτελεί bottleneck.

### Σύστημα Μνήμης

Ο αναγνώστης DRAM έχει διεπαφή με το υποσύστημα Network on Chip και ευθύνεται για την διαχείριση των αιτημάτων για μνήμη από άλλα συστήματα. Αυτό το σύστημα περιέχει τον προεπεξεργαστή και την λογική buffer πλαισίων για αποσυμπίεση. Ο εγγραφές DRAM επίσης έχει διεπαφή με το Network on Chip.

Το στάδιο που απαιτεί την μεγαλύτερη χρήση μνήμης είναι η εκτίμηση κίνησης ώστε να βρεθούν τμήματα εικονοστοιχεία από πλαίσια που είναι όμοια με το τωρινό πλαίσιο. Γι' αυτό ένα σημαντικό κομμάτι της αρχιτεκτονικής είναι μια αποθήκη αναφοράς πίνακα που αποθηκεύει το παράθυρο για εύρεση κίνησης. Αυτή η αποθήκη χρησιμοποιεί αλγόριθμο λιγότερο πρόσφατα χρησιμοποιημένου τμήματος (Least Recently Used - LRU) για να διαγράψει εγγραφές.

Το σύστημα μνήμης μπορεί να προβλέψει αποδοτικά τα τμήματα που θα χρειαστούν που σημαίνει ότι είναι ανθεκτικό σε υψηλή καθυστέρηση μνήμης και υψηλή παραλληλοποίηση μνήμης.

### Έλεγχος και Λειτουργία Χωρίς Κατάσταση

Ο κωδικοποιητής προγραμματίζεται από καταχωρητές ελέγχου και κατάστασης για κάθε διεργασία. Όλες οι εισοδοί και οι εξοδοί αποθηκεύονται στο VCU DRAM, αυτό επιτρέπει την αλλαγή VCU κατά την διάρκεια εκτέλεσης καθώς η DRAM είναι εξωτερικά από το ολοκληρωμένο.

### 3.1.3 Σχεδιασμός Λογισμικού και Υλικού

Όλα τα κομμάτια του συστήματος εκτός από το ολοκληρωμένο ASIC είναι αγορασμένα και είναι εγκατεστημένα πάνω σε κάρτες PCI, κάθε σύστημα περιέχει 5 VCU κάρτες και κάθε κάρτα VCU περιέχει 2 VCU.

### Μνήμη

Ο κάθε πυρήνας επεξεργασίας μπορεί να κωδικοποιήσει βίντεο 2160p σε πραγματικό χρόνο με μέχρι και 60 εικόνες το δευτερόλεπτο (FPS). Το κάθε πλαίσιο είναι 11.9 MiB με το μέσο εύρος ζώνης 3.5 GiB ανά δευτερόλεπτο. Το εύρος ζώνης στη χειρότερη περίπτωση είναι περίπου 3GiB και την μέση περίπτωση 2.2GiB, άρα εφόσον ένα μηχάνημα έχει 10 πυρήνες τότε χρειάζεται περίπου 27 με 37 GiB εύρους ζώνης μνήμης που παρέχονται μέσω τεσσάρων 32b LPDDR4-3200 κανάλια με συνολικό εύρος ζώνης περίπου 36 GiB ανά δευτερόλεπτο.

### Firmware

Το Firmware υπάρχει σε ολοκληρωμένο στη κάρτα PCI, αυτό χρονοπρογραμματίζει τις διεργασίες που θα ολοκληρωθούν στο σύστημα. Το σύστημα είναι σχεδιασμένο με την υπόθεση ότι



θα εκτελεστούν πολλά προγράμματα χρήστη ώστε να βελτιστοποιηθεί η χρήση του και έτσι το VCU είναι ικανό να χειριστεί πολλαπλές ροές.

### Σχεδιασμός Χρονοπρογραμματισμού

Ο χρονοπρογραμματιστής που χρησιμοποιείται είναι ένας επιγραμμικός πολυδιάστατος που χρησιμοποιεί την πακετοποίηση σε κάδους (bin packaging). Αυτό σημαίνει ότι κανένας πυρήνας δεν υπερφορτίζεται με δουλειές και καμία δουλειά δεν ξεμένει από πόρους. Η κάθε συστάδα έχει πολλά λογικά αποθέματα από πόρους που χαρακτηρίζονται από περίπτωση χρήσης και πρωτεραιότητα, τα αποθέματα μπορούν να αλλάζουν πόρους ανάλογα με τις απαιτήσεις. Το κάθε απόθεμα έχει δικό του χρονοπρογραμματιστή και διαφορετικούς εργάτες διαφορετικών τύπων. Επίσης μερικά αποθέματα έχουν δικό τους VCU και μπορεί να έχουν και κανονική CPU επεξεργασία.

Οι χρονοπρογραμματιστές είναι σχεδιασμένοι για κλιμάκωση, δηλαδή να διατηρούν μικρή καθυστέρηση (latency) και παράλληλα να έχουν πολλούς εργάτες. Η εργασία κατανεμείτε σύμφωνα με έναν άπληστο αλγόριθμο για την μεγιστοποίηση του φόρτου, αυτό σημαίνει ότι ορισμένοι πόροι μπορεί να μείνουν αχρησιμοποίητοι, να αφαιρεθούν από το αποθέμα και να προστεθούν σε άλλο.

Στην περίπτωση σφάλματος κατά την κωδικοποίηση η κωδικοποίηση γίνεται σε άλλο πυρήνα ή με τη χρήση λογισμικού.

### 3.1.4 Σύνθεση Υψηλού Επιπέδου

Ο σχεδιασμός του υλικού έγινε με την πλατφόρμα Catapult της Mentor Graphics και με τη χρήση ενός εσωτερικού λογισμικού εν ονόματι Tuffel. Ο σχεδιασμός των ροών δεδομένων έχει γίνει χρησιμοποιώντας μια Σύνθεση Υψηλού Επιπέδου που βασίζεται στην C++, αυτό επιτρέπει την χρήση τεχνικών για την εύρεση προβλημάτων στο υλικό καθώς και μειώνει τον κώδικα που πρέπει να γραφεί. Στο επίπεδο ASIC είναι ευκολότερο να υλοποιηθούν μηχανισμοί για την αντοχή σε σφάλματα οι οποίοι βασίζονται στην περιεκτικότητα και την διαχείριση σφαλμάτων από το λογισμικό.

## 3.2 Απόδοση

Σε αυτή την ενότητα αναλύεται η απόδοση του συστήματος κωδικοποίηση Βίντεο.

### Εκτίμηση Απόδοσης και Ποιότητας

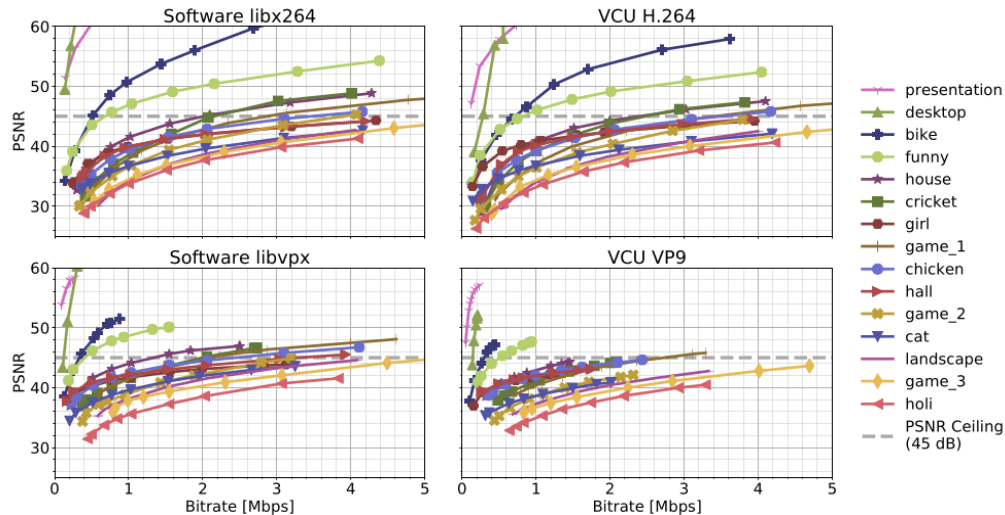
Το πρόγραμμα που χρησιμοποιήθηκε για την εκτίμηση της απόδοσης είναι το vbench που περιέχει 15 βίντεο που είναι οργανωμένα ανάλογα με την ανάλυση τους, τον ρυθμό εικόνων και την εντροπία. Η κωδικοποίηση γίνεται με το πρόγραμμα ffmpeg και μετريέται η παραγωγικότητα σε εικονοστοιχεία ανά δευτερόλεπτο.

Τα συστήματα που θα εξεταστούν για σύγκριση με τα VCUs είναι ένας εξυπηρετητής με δύο επεξεργαστές Skylake και 384 GiB μνήμης DRAM και ένα σύστημα με 4 κάρτες γραφικών Nvidia T4. Όσο αφορά τα VCU θα εξεταστεί ένα σύστημα με 4, δηλαδή με 8 πυρήνες και ένα σύστημα με 10, δηλαδή με 20 πυρήνες.

Στον πίνακα 3.1 φαίνεται πως τα VCU είναι πολύ πιο αποδοτικό από τα υπάρχοντα συστήματα όσο αφορά την κωδικοποίηση. Το Perf/TCO συμβολίζει την απόδοση σχετικά με το συνολικό κόστος ανάλογα με το κόστος της ιδιοκτησίας. Επίσης φαίνεται ότι η κωδικοποίηση VP9 έχει πολύ μεγάλη βελτίωση σε σχέση με την υλοποίηση του επεξεργαστή.

System	Throughput (Mpix/s)		Perf/TCO	
	H.264	VP9	H.264	VP9
Skylake	714	154	1.0x	1.0x
4xNvidia T4	2.484	-	1.5x	-
8xVCU	5.973	6.122	4.4x	20.8x
20xVCU	14.932	15.306	7.0x	33.3x

Table 3.1: Σύγκριση Απόδοσης Συστημάτων



Εικόνα 3.1: Σχέση PSNR με Bitrate Κωδικοποίησης

Πέρα από την απόδοση κωδικοποίησης σημαντική είναι και η ενεργειακή απόδοση του συστήματος όπου πάλι υπάρχει βελτίωση της τάξης 6.7x στο H.264 και 68.9x στο VP9.

**Συμπερασματικά** οι μεγάλες βελτιώσεις ήταν αναμενόμενες καθώς ο πυρήνας κωδικοποίησης του VCU είναι υλοποιημένος σε ASIC, που όπως υπόθηκε στο πρώτο κεφάλαιο (1.2.1) είναι πολύ πιο αποδοτικό όσο αφορά την απόδοση και την ενεργειακή κατανάλωση από του κανονικούς επεξεργαστές.

### Σηματοθορυβική Απόδοση

Η σηματοθορυβική απόδοση (απόδοση PSNR) μετρήθηκε με τα βίντεο του vbench από την κωδικοποίηση με ffmpeg.

Στην εικόνα 3.1 φαίνεται η σχέση μέγιστου σηματοθορυβικού λόγου (PSNR) με ταχύτητα (bitrate) για κάθε βίντεο που υπάρχει στο vbench. Στο VP9 φαίνεται πως υπάρχει διαφορά στο Bitrate, δηλαδή το VCU κάνει καλύτερη συμπίεση στα δεδομένα διατηρώντας καλή ποιότητα δεδομένων. Τα βίντεο που είναι ταξινομημένα χαμηλότερα από τα άλλα έχουν υψηλότερη κίνηση, δηλαδή μεγαλύτερη εντροπία. Τα βίντεο που περιέχει η σουίτα vbench είναι [20]:

Η εντροπία του βίντεο πολύ συνοπτικά δείχνει το πόσο αλλάζει το βίντεο όσο αναπαραγάγεται, όσο μεγαλύτερη τόσο περισσότερο μέγεθος θα έχει συμπίεσμένη.

Ανάλυση	Όνομα	Εντροπία
854x480	cat	6.8
854x480	holi	7.0
1280x720	desktop	0.2
1280x720	bike	0.9
1280x720	cricket	3.4
1280x720	game2	4.9
1280x720	girl	5.9
1280x720	game3	6.1
1920x1080	presentation	0.2
1920x1080	funny	2.5
1920x1080	house	3.6
1920x1080	game1	4.6
1920x1080	landscape	7.2
1920x1080	hall	7.7
3840x2160	chicken	5.9

Table 3.2: Ανάλυση και Εντροπία ανά βίντεο

Από τον πίνακα 3.2 και την εικόνα 3.1 μπορούμε να συμπεράνουμε ότι στα βίντεο πολύ χαμηλής εντροπίας (με εντροπία το πολύ 1, δηλαδή τα βίντεο presentation, desktop και bike) το σύστημα μειώνει αρκετά το μέγεθος του αρχείου. Τα βίντεο με μέση εντροπία ([2.5 - 4.9]) στην κωδικοποίηση H.264 δεν έχουν μεγάλη διαφορά και στην κωδικοποίηση VP9 έχουν μικρότερο μέγεθος αρχείου. Τέλος τα βίντεο με υψηλή εντροπία ([5.0 - 10.0]) πάλι στο H.264 δεν υπάρχει σημαντική διαφορά αλλά στο VP9 υπάρχει μείωση στον απαιτούμενο χώρο για αποθήκευση.

### 3.3 Συμπέρασμα

Το σύστημα αυτό φτιάχτηκε για την βελτίωση της κωδικοποίησης για το Youtube και για λοιπές λειτουργίες που χρειάζεται η Google και όπως φαίνεται από την απόδοση του επιτυγχάνει τον σκοπό του. Αυτή η επιτυχία δίνει ότι τα ενσωματωμένα συστήματα μεγάλης κλίμακας ενδέχεται να υιοθετηθούν και για άλλες χρήσεις ώστε να βελτιωθεί η απόδοση. Αξίζει να σημειωθεί ότι αυτή είναι η πρώτη φορά που γίνεται κάτι τέτοιο και ενδέχεται να μην είναι βελτιστοποιημένο ως προς την απόδοση.

# Ακρονύμια

LUT    Look Up Table  
VCU    Video Coding Unit



# Αντιστοίχιση Ελληνικών και Αγγλικών Όρων

Πολυμέσα	Multimedia
Ενσωματωμένα Συστήματα	Embedded Systems
Πραγματικού Χρόνου	Real Time
Υλικό	Hardware
Κωδικοποίηση	Encoding
Αποκωδικοποίηση	Decoding
Μετατροπή	Transcoding
Δίαυλοι	Buses
Είσοδος	Input
Έξοδος	Output
Αισθητήρας	Sensor
Εικονοστοιχείο	Pixel
Συνιστώσα	Coefficient
Μεταβλητή	Variable
Ιδιόκτητος	Proprietary
Επιγραμμικός	Online
Περικτικότητα	Redundancy



# Βιβλιογραφία

- [1] John L. Hennessy and David A. Patterson. “A New Golden Age for Computer Architecture”. In: *Commun. ACM* 62.2 (Jan. 2019), pp. 48–60. ISSN: 0001-0782. DOI: 10.1145/3282307. URL: <https://doi.org/10.1145/3282307>.
- [2] Cam Cullen. “Sandvine Internet Phenomena Report Q3 2019”. In: (2019).
- [3] Peter Marwedel. *Embedded System Design*. Springer International Publishing, 2021.
- [4] David J Katz and Rick Gentile. *Embedded media processing*. Elsevier, 2005.
- [5] Μηνάς Δασυγένης και Δημήτριος Σούντρης. *Ενσωματωμένα Συστήματα*. Κάλλιπος, 2015.
- [6] Wayne Hendrix Wolf. *Computers as Components: Principles of Embedded Computing System Design*. 2001.
- [7] Taoufik Saidani, Mohamed Atri, Lazhar Khriji, and Rached Tourki. “An efficient hardware implementation of parallel EBCOT algorithm for JPEG 2000”. In: *Journal of Real-Time Image Processing* 11.1 (2016), pp. 63–74.
- [8] Arthur Schuchter and Andreas Uhl. “Embedded hardware low cost JPEG 2000 video coding system: Hardware coder for surveillance type videos”. In: *J. Real-Time Image Processing* 5 (Sept. 2010), pp. 149–162. DOI: 10.1007/s11554-009-0131-3.
- [9] P. Havaladar and G. Medioni. *Multimedia Systems: Algorithms, Standards, and Industry Practices*. Cengage Learning, 2009. ISBN: 9781418835941. URL: <https://books.google.gr/books?id=k83m5inkjXwC>.
- [10] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Pearson Education, 2011. ISBN: 9780133002324.
- [11] Z-e. Baarir. *Entropy Encoding EBCOT (Embedded Block Coding with Optimized Truncation) In JPEG2000*. 2011.
- [12] Tokuhiko Tsukiyama, Yoshie Katsuharu, Shinpei Saba, Syoji Ozaki, and Kunihiro Itoh. *Method and system for data compression and restoration*. 1983.
- [13] *Spartan-3A DSP FPGA Family Data Sheet*. URL: [https://www.xilinx.com/support/documentation/data\\_sheets/ds610.pdf](https://www.xilinx.com/support/documentation/data_sheets/ds610.pdf).
- [14] Manjunath Gangadhar and Dinesh Bhatia. “FPGA based EBCOT architecture for JPEG 2000”. In: *Microprocessors and Microsystems* 29.8 (2005). Special Issue on FPGAs: Case Studies in Computer Vision and Image Processing, pp. 363–373. ISSN: 0141-9331. DOI: <https://doi.org/10.1016/j.micpro.2004.10.006>. URL: <https://www.sciencedirect.com/science/article/pii/S0141933104001449>.
- [15] Kishor Sarawadekar and Swapna Banerjee. “A High Speed bit Plane Coder for JPEG2000 and it's FPGA Implementation”. In: (2009).
- [16] *ADV202 Data Sheet*. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADV202.pdf>.



- [17] *ADSP-BF533 Blackfin® Processor Hardware Reference*. 2013. URL: [https://www.analog.com/media/en/dsp-documentation/processor-manuals/ADSP-BF533\\_hwr\\_rev3.6.pdf](https://www.analog.com/media/en/dsp-documentation/processor-manuals/ADSP-BF533_hwr_rev3.6.pdf).
- [18] *Reimagining video infrastructure to empower YouTube*. 2021. URL: <https://blog.youtube/inside-youtube/new-era-video-infrastructure>.
- [19] Parthasarathy Ranganathan, Daniel Stodolsky, Jeff Calow, Jeremy Dorfman, Marisabel Guevara, Clinton Wills Smullen IV, Aki Kuusela, Raghu Balasubramanian, Sandeep Bhatia, Prakash Chauhan, et al. "Warehouse-scale video acceleration: co-design and deployment in the wild". In: *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 2021, pp. 600–615.
- [20] Andrea Lottarini, Alex Ramirez, Joel Coburn, Martha A. Kim, Parthasarathy Ranganathan, Daniel Stodolsky, and Mark Wachsler. "Vbench: Benchmarking Video Transcoding in the Cloud". In: *SIGPLAN Not.* 53.2 (Mar. 2018), pp. 797–809. ISSN: 0362-1340. DOI: 10.1145/3296957.3173207. URL: <https://doi.org/10.1145/3296957.3173207>.