



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ  
ΤΟΜΕΑΣ ΥΛΙΚΟΥ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

**Σχεδίαση και υλοποίηση του υλικού της  
διεπαφής ενός οπτικού αισθητήρα σε System On Chip  
για την επεξεργασία εικόνας και τη συμπίεση video**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Γεώργιος Καρακωνσταντής**

**Επιβλέπων :** Γεώργιος Σταμούλης  
Αναπληρωτής Καθηγητής

Βόλος, Ιούλιος 2005

Copyright © Γεώργιος Καρακωνσταντής, 2005  
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Η εργασία αυτή είναι αφιερωμένη  
στην οικογένειά μου.

Καταρχήν θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Γ. Σταμούλη για την υποστήριξη του κατά την διάρκεια της εργασίας μου. Η καθοδήγηση και οι συμβουλές του, αποτέλεσαν καθοριστικό παράγοντα της πορείας και εξέλιξης τόσο των σπουδών μου όσο και της σταδιοδρομίας μου. Τον ευχαριστώ για την εμπιστοσύνη που μου δείχνει και τις σημαντικές ευκαιρίες που μου έδωσε.

Επίσης, ευχαριστώ το μέλος της εξεταστικής επιτροπής της παρούσας διπλωματικής εργασίας κ. Ν. Ευμορφόπουλο για τις εύστοχες παρατηρήσεις του κειμένου.

Ένα μεγάλο ευχαριστώ στον κ. Ν. Μπέλλα, ερευνητή στα εργαστήρια της Motorola στο Σικάγο των Ηνωμένων Πολιτειών, για τη παροχή βοήθειας και την επίλυση προβλημάτων που αντιμετώπισα κατά τη διάρκεια της εργασίας μου, μέσα από τις πολύωρες υπερατλαντικές τηλεφωνικές συνομιλίες. Οι συμβουλές του καθόρισαν την ανάπτυξη και ολοκλήρωση της εργασίας. Τον ευχαριστώ για τις εύστοχες συμβουλές και την βοήθεια στην εξέλιξη της σταδιοδρομίας μου.

Ευχαριστώ το φίλο και συνάδελφο κ. Δ. Καραμπατζάκη για την βοήθεια και καθοδήγηση στην επίλυση προβλημάτων που αντιμετώπισα όλο αυτό το διάστημα και την υπόλοιπη ομάδα του γραφείου E5 για την υποστήριξη και τη παρέα.

Επίσης θα ήθελα να ευχαριστήσω τους φίλους και συμφοιτητές μου για την κατανόηση και τη παρέα τους όλο αυτό το διάστημα.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου για την συμπαράσταση και την υποστήριξη των αποφάσεων μου σε όλη τη διάρκεια των σπουδών μου.

## Περιεχόμενα

<b>Εισαγωγή</b> .....	1
Οργάνωση της διπλωματικής .....	4
<b>1. Ψηφιακή επεξεργασία εικόνας</b> .....	6
1.1 Εισαγωγή .....	7
1.2 Η αλυσίδα επεξεργασίας εικόνας .....	9
1.3 Καταγραφή εικόνας .....	11
1.4 Στάδια αλυσίδας επεξεργασίας .....	15
<b>2. Αλγόριθμοι προ - επεξεργασίας</b> .....	21
2.1 Εισαγωγή .....	22
2.2 Αλγόριθμος εξισορρόπησης του λευκού (White Balance - AWB) .....	22
2.2.1 Τεχνικές .....	23
2.2.2 Υπόθεση Gray World .....	25
2.2.3 Υλοποίηση .....	26
2.3 Αλγόριθμος αυτόματης έκθεσης (Auto Exposure - AE) .....	31
2.3.1 Τεχνικές .....	31
2.3.2 Υλοποίηση .....	33
<b>3. Οπτικοί Αισθητήρες</b> .....	37
3.1 Εισαγωγή .....	38
3.2 Διαφορές .....	41
3.3 Οπτικοί αισθητήρες CMOS .....	44
3.4 Ο αισθητήρας επιλογής μας .....	49
3.4.1 Γενικά Χαρακτηριστικά .....	49
3.4.2 Αρχιτεκτονική .....	50
3.4.3 Έλεγχος κάμερας .....	51
3.5 Συμπεράσματα .....	54
<b>4. Πρωτόκολλο Επικοινωνίας I<sup>2</sup>C (Inter-Integrated Circuit)</b> .....	55
4.1 Εισαγωγή .....	56
4.2 Χαρακτηριστικά .....	56
4.3 Μεταφορά δεδομένων .....	58
4.4 Ρολόι .....	61
4.5 Συμπεράσματα .....	61
<b>5. Ροή πληροφορίας σχεδίασης</b> .....	62
5.1 Γενική ροή πληροφορίας σχεδίασης VLSI .....	63
5.2 Ροή πληροφορίας σχεδίασης .....	65
5.3 Γλώσσα περιγραφής υλικού VHDL .....	67
5.4 Διάταξη προγραμματιζόμενης λογικής - FPGA .....	68
5.5 FPGA σχεδίασης .....	72

<b>6. Αρχιτεκτονική Συστήματος</b> .....	75
6.1 Εισαγωγή .....	76
6.2 Πρωτόκολλο επικοινωνίας I <sup>2</sup> C.....	76
6.2.1 Παραγωγή scl, sda start και stop σημάτων .....	80
6.3 Καταγραφή εικόνας .....	83
6.4 Μέτρηση των χρωματικών συνιστωσών .....	86
6.5 Σύστημα μνήμης .....	87
6.6 Κύκλωμα ελέγχου συστήματος.....	87
6.7 Έλεγχος καταχωρητών .....	89
<b>7. Περιγραφή γλώσσας υλικού - Λειτουργική προσομοίωση</b> .....	90
7.1 Εργαλεία .....	91
7.2 Πρωτόκολλο επικοινωνίας I <sup>2</sup> C.....	92
7.3 Σύλληψη εικόνας.....	98
7.4 Μέτρηση των χρωμάτων .....	101
7.5 Σύστημα μνήμης .....	102
7.5.1 Κύκλωμα ελέγχου.....	103
7.5.2 Στοιχείο μνήμης.....	104
7.6 Κύκλωμα ελέγχου συστήματος.....	106
7.6.1 Κύκλωμα διασύνδεσης.....	108
7.7 Ανάγνωση εικόνας .....	110
7.8 Το συνολικό κύκλωμα .....	112
<b>8. Σύνθεση και Υλοποίηση</b> .....	115
8.1 Ροή σχεδίασης - εργαλεία.....	116
8.2 Πρωτόκολλο επικοινωνίας I <sup>2</sup> C.....	119
8.3 Καταγραφή εικόνας .....	124
8.4 Σύστημα μνήμης .....	127
8.5 Συνολικό Κύκλωμα.....	130
8.5.1 Κύκλωμα ελέγχου συστήματος.....	130
8.5.2 Κύκλωμα διασύνδεσης.....	134
<b>9. Συμπεράσματα Άξονες Μελλοντικής Ανάπτυξης</b> .....	139
9.1 Συμπεράσματα.....	139
9.2 Μελλοντική εργασία.....	141
<b>Παράρτημα</b> .....	144
Πρωτόκολλο επικοινωνίας I <sup>2</sup> C.....	144
Καταγραφή εικόνας.....	145
Σύστημα μνήμης.....	147
Συνολικό Κύκλωμα .....	149
<b>Βιβλιογραφία</b> .....	150

## Κατάλογος σχημάτων

Σχήμα E.1: Σύστημα πολυμέσων ενσωματωμένο σε μια FPGA.....	3
Σχήμα 1.1: Πίνακας pixel εικόνας.....	8
Σχήμα 1.2: Τυπική αλυσίδα προεπεξεργασίας εικόνας .....	10
Σχήμα 1.3: Κλίμακα του γκρι.....	11
Σχήμα 1.4: Φάσμα ευαισθησίας χρωματικών φίλτρων [8].....	12
Σχήμα 1.5: Πρωταρχικά χρώματα .....	13
Σχήμα 1.6: Συμπληρωματικά χρώματα.....	13
Σχήμα 1.7: Καταγραφή ασπρόμαυρης εικόνας .....	14
Σχήμα 1.8: Καταγραφή έγχρωμης εικόνας - Bayer pattern .....	14
Σχήμα 1.9: Χρήση φακών - Bayer pattern [12].....	14
Σχήμα 1.10: Interlaced σάρωση.....	15
Σχήμα 1.11: Progressive σάρωση.....	15
Σχήμα 1.12: Color Interpolation -Bayer format .....	17
Σχήμα 1.13: Εικόνα πριν και μετά την εφαρμογή interpolation .....	17
Σχήμα 1.14: Τεχνική color interpolation.....	18
Σχήμα 1.15: Βελτίωση εικόνας με εφαρμογή αλυσίδας επεξεργασίας.....	20
Σχήμα 2.1: Αλγόριθμοι προ επεξεργασίας εικόνας.....	22
Σχήμα 2.2: Είδη φωτισμού καταγραφής .....	22
Σχήμα 2.3: Εφαρμογή αλγορίθμου εξισορρόπησης λευκού.....	25
Σχήμα 2.4: Εφαρμογή υπόθεσης Gray World .....	26
Σχήμα 2.5: Εφαρμογή white balance .....	28
Σχήμα 2.6: (α) Αρχική εικόνα, (β) Εφαρμογή εξισορρόπησης λευκού .....	28
Σχήμα 2.7: Εφαρμογή εξισορρόπησης λευκού στην δημοφιλή εικόνα Lena.....	29
Σχήμα 2.8: (α) Αρχική εικόνα, (β) Εφαρμογή white balance.....	30
Σχήμα 2.9: (α) Αρχική εικόνα, (β) Εφαρμογή εξισορρόπησης λευκού .....	30
Σχήμα 2.10: Αλγόριθμος - εικόνα χωρισμένη σε 2 τμήματα.....	32
Σχήμα 2.11: Επίπεδα έκθεσης .....	33
Σχήμα 3.1: Εφαρμογές οπτικών αισθητήρων .....	38
Σχήμα 3.2: Αισθητήρας και photosites .....	41
Σχήμα 3.3: Αποθήκευση ηλεκτρονίων.....	42
Σχήμα 3.4: Σχηματικό Block διάγραμμα μιας έγχρωμης Video κάμερας.....	42
Σχήμα 3.5: CMOS αισθητήρας.....	44
Σχήμα 3.6 Αρχιτεκτονική CMOS αισθητήρα.....	45
Σχήμα 3.7: (α) Φίλτρο Bayer, (β) Mosaic Array, (γ) Φωτοδίοδοι .....	45
Σχήμα 3.8: Ενεργό pixel του αισθητήρα.....	45
Σχήμα 3.9: Σύστημα οπτικού αισθητήρα CMOS.....	47
Σχήμα 3.10: Fill factor .....	48
Σχήμα 3.11: Λειτουργικό Block Διάγραμμα .....	50
Σχήμα 3.12: Πίνακας pixel .....	53
Σχήμα 3.13: Σχέδιο χρωμάτων pixel.....	53
Σχήμα 3.14: Vertical timing.....	53
Σχήμα 3.15: Horizontal timing .....	53
Σχήμα 4.1: Απλή υλοποίηση I <sup>2</sup> C.....	57
Σχήμα 4.2: Start - Stop σήμα .....	57
Σχήμα 4.3: Μεταφορά της διεύθυνσης της slave συσκευής [48].....	59

## Περιεχόμενα

---

Σχήμα 4.4: Λήψη δεδομένων απο τη slave συσκευή .....	60
Σχήμα 4.5: Μεταφορά δεδομένων.....	60
Σχήμα 4.6: Διαδικασία εγγραφής καταχωρητή R0x09:0—Value 0x0284 .....	60
Σχήμα 4.7: Διαδικασία ανάγνωσης καταχωρητή R0x09:0 .....	61
Σχήμα 5.1: Ροή πληροφορίας VLSI [47] .....	63
Σχήμα 5.2: Σχεδίαση ολοκληρωμένου κυκλώματος .....	64
Σχήμα 5.3: Ροή πληροφορίας FPGA [40].....	65
Σχήμα 5.4: Place And Route .....	67
Σχήμα 5.5: Γενική δομή μιας διάταξης FPGA .....	69
Σχήμα 5.6: Λογική βαθμίδα επαναπρογραμματιζόμενων FPGA .....	69
Σχήμα 5.7: Look Up Table.....	70
Σχήμα 5.8: LUT και flip flops .....	70
Σχήμα 5.9: Λογική βαθμίδα OTP (One Time Programmable) FPGA .....	70
Σχήμα 5.10: Διακόπτες FPGA.....	71
Σχήμα 5.11: Εγκατάσταση σχεδίασης.....	71
Σχήμα 5.12: Spartan ΙΙΕ .....	72
Σχήμα 5.13: Αρχιτεκτονική Spartan ΙΙΕ [37] .....	72
Σχήμα 5.14: Block διάγραμμα Spartan ΙΙΕ.....	73
Σχήμα 5.15: Τμήμα CLB, 2 λογικά κελιά.....	74
Σχήμα 5.16: Board με FPGA [43] .....	74
Σχήμα 6.1: Αρχιτεκτονική συστήματος .....	76
Σχήμα 6.2: I <sup>2</sup> C Πρωτόκολλο επικοινωνίας.....	77
Σχήμα 6.3: I <sup>2</sup> C Πρωτόκολλο επικοινωνίας με πρόσθετα χαρακτηριστικά.....	79
Σχήμα 6.4: Μηχανή σχεδιασμένη στο HDL Designer.....	80
Σχήμα 6.5: Μηχανή πεπερασμένων καταστάσεων παραγωγής scl και sda σημάτων.....	81
Σχήμα 6.6: Μηχανή πεπερασμένων καταστάσεων καταγραφής εικόνας .....	84
Σχήμα 6.7: Μηχανή πεπερασμένων καταστάσεων παραγόμενη από το HDL Designer. 86	
Σχήμα 6.8: Μηχανή πεπερασμένων καταστάσεων κυκλώματος ελέγχου .....	88
Σχήμα 7.1: Συνολικό κύκλωμα πρωτοκόλλου επικοινωνίας I <sup>2</sup> C .....	92
Σχήμα 7.2: Block διάγραμμα κυκλώματος πρωτοκόλλου επικοινωνίας I <sup>2</sup> C.....	93
Σχήμα 7.3: (α) Κύκλωμα ολισθητή, (β) Κύκλωμα μετρητή .....	94
Σχήμα 7.4: Αποτελέσματα προσομοίωσης αποστολής δεδομένων .....	96
Σχήμα 7.5: Αποτελέσματα προσομοίωσης αποστολής δεδομένων παραθύρου list.....	97
Σχήμα 7.6: Λήψη δεδομένων.....	97
Σχήμα 7.7: Block διάγραμμα κυκλώματος καταγραφής εικόνας.....	99
Σχήμα 7.8: Καταγραφή εικόνας .....	100
Σχήμα 7.9: Ολοκλήρωση λήψης πλαισίου.....	100
Σχήμα 7.10: Block διάγραμμα κυκλώματος μέτρησης χρωμάτων.....	101
Σχήμα 7.11: Σύστημα μνήμης .....	102
Σχήμα 7.12: Block διάγραμμα κυκλώματος ελέγχου της μνήμης.....	103
Σχήμα 7.13: Block διάγραμμα στοιχείου μνήμης .....	104
Σχήμα 7.14: Μνήμη γεμάτη .....	105
Σχήμα 7.15: Ολοκλήρωση ανάγνωσης μνήμης .....	105
Σχήμα 7.16: Ολοκλήρωση εγγραφής πλαισίου και ανάγνωσης μνήμης.....	105
Σχήμα 7.17: Κύκλωμα ελέγχου συστήματος.....	106
Σχήμα 7.18: Block διάγραμμα κύκλωμα ελέγχου συστήματος.....	106
Σχήμα 7.19: Αρχικοποίηση I <sup>2</sup> C .....	107
Σχήμα 7.20: Ολοκλήρωση λήψης εικόνας, ανάγνωση από το σύστημα .....	108

---



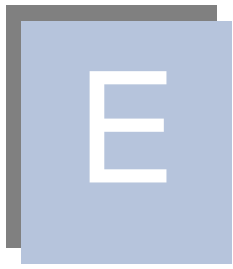
## Περιεχόμενα

---

Σχήμα 7.21: Κύκλωμα διασύνδεσης καταγραφής εικόνας και συστήματος μνήμης.....	108
Σχήμα 7.22: Εγγραφή και ανάγνωση εικόνας .....	109
Σχήμα 7.23: Ολοκλήρωση ανάγνωσης .....	110
Σχήμα 7.24: Block διάγραμμα κυκλώματος ανάγνωσης εικόνας .....	110
Σχήμα 7.25: Κύκλωμα ανάγνωσης εικόνας.....	111
Σχήμα 7.26: Ανάγνωση εικόνας .....	112
Σχήμα 7.27: Συνολικό κύκλωμα.....	112
Σχήμα 7.28: Έναρξη λήψης εικόνας .....	113
Σχήμα 7.29: Ολοκλήρωση λήψης πλαισίου.....	113
Σχήμα 7.30: Τέλος γραμμής πλαισίου .....	114
Σχήμα 7.31: Ανάγνωση μνήμης.....	114
Σχήμα 8.1: Ροή σχεδίασης εργαλείου ISE 7.1i.....	116
Σχήμα 8.2: Block διάγραμμα πρωτοκόλλου.....	119
Σχήμα 8.3: Ολισθητής κυκλώματος πρωτοκόλλου .....	120
Σχήμα 8.4: Επίπεδο πυλών .....	120
Σχήμα 8.5: Τοποθέτηση πρωτοκόλλου .....	123
Σχήμα 8.6: Τοποθέτηση κυκλώματος στο FPGA.....	123
Σχήμα 8.7: Σύγκριση κυματομορφών λειτουργικής και Post Map προσομοίωσης .....	124
Σχήμα 8.8: Κατανάλωση ισχύος.....	124
Σχήμα 8.9: Block διάγραμμα κυκλώματος καταγραφής εικόνας.....	125
Σχήμα 8.10: Συνοπτικά αποτελέσματα υλοποίησης σχεδίασης .....	126
Σχήμα 8.11: Τοποθέτηση κυκλώματος .....	127
Σχήμα 8.12: Αποτελέσματα σύνθεσης - Block διάγραμμα κυκλώματος μνήμης .....	128
Σχήμα 8.13: Αποτελέσματα σύνθεσης - Block διάγραμμα στοιχείου μνήμης.....	128
Σχήμα 8.14: Τοποθέτηση κυκλώματος στο FPGA (εργαλείο Floorplanner) .....	130
Σχήμα 8.15: Αποτελέσματα σύνθεσης - Block διάγραμμα κυκλώματος ελέγχου .....	130
Σχήμα 8.16: Τοποθέτηση κυκλώματος (εργαλείο Floorplanner).....	132
Σχήμα 8.17: Δίκτυο σήματος reset.....	133
Σχήμα 8.18: Τοποθέτηση κυκλώματος στο FPGA (εργαλείο FPGA editor) .....	133
Σχήμα 8.19: Κατανάλωση ισχύος.....	134
Σχήμα 8.20: Αποτελέσματα σύνθεσης - Block διάγραμμα κυκλώματος διασύνδεσης ..	134
Σχήμα 8.21: Μέτρηση χρωματικών συνιστωσών.....	135
Σχήμα 8.22: Μετρητής.....	135
Σχήμα 8.23: Καθυστερήση μονοπατιού - Post Place And Route Static Timing .....	137
Σχήμα 8.24: Τοποθέτηση κυκλώματος στο FPGA (εργαλείο Floorplanner) .....	137

## Κατάλογος πινάκων

Πίνακας 2.1: Έλεγχος εφαρμογής αλγορίθμου αυτόματης έκθεσης.....	34
Πίνακας 2.2: Εύρεση νέου χρόνου έκθεσης σε ψευδογλώσσα .....	35
Πίνακας 3.1: Έξοδος σε YCbCr μορφή.....	51
Πίνακας 3.2: Έξοδος σε επεξεργασμένη Bayer Μορφή.....	52
Πίνακας 3.3: Έξοδος σε RGB μορφή.....	52
Πίνακας 6.1: Περιγραφή καταστάσεων του I <sup>2</sup> C Πρωτοκόλλου επικοινωνίας.....	78
Πίνακας 6.2: Περιγραφή καταστάσεων της μηχανής παραγωγής scl και sda σημάτων...	82
Πίνακας 6.3: Περιγραφή καταστάσεων κυκλώματος καταγραφής εικόνας.....	85
Πίνακας 6.4: Περιγραφή καταστάσεων κυκλώματος ελέγχου.....	89
Πίνακας 7.1: Περιγραφή πυλών κυκλώματος πρωτοκόλλου επικοινωνίας I <sup>2</sup> C.....	94
Πίνακας 7.2: Παράδειγμα .do αρχείου για προσομοίωση αποστολής δεδομένων.....	98
Πίνακας 7.3: Περιγραφή πυλών κυκλώματος .....	99
Πίνακας 7.4: : Παράδειγμα .do αρχείου για προσωμείωση καταγραφής εικόνας .....	101
Πίνακας 7.5: Περιγραφή πυλών κυκλώματος μέτρησης χρωμάτων .....	102
Πίνακας 7.6: Περιγραφή πυλών κυκλώματος ελέγχου της μνήμης.....	103
Πίνακας 7.7: Περιγραφή πυλών κυκλώματος ελέγχου συστήματος.....	107
Πίνακας 7.8: Περιγραφή πυλών κυκλώματος διασύνδεσης .....	109
Πίνακας 8.1: Αναφορά σύνθεσης.....	121
Πίνακας 8.2: Αναφορά MAP.....	121
Πίνακας 8.3: Αναφορά Place And Route (χρήση στοιχείων).....	121
Πίνακας 8.4: Αναφορά Place And Route (καθυστερήσεις (ns)) .....	122
Πίνακας 8.5: Post Place And Route Static Timing .....	122
Πίνακας 8.6: Χρήση πόρων συσκευής.....	125
Πίνακας 8.7: Αποτελέσματα Place And Route.....	126
Πίνακας 8.8: Αποτελέσματα Post Place And Route Static Timing .....	127
Πίνακας 8.9: Αναφορά σύνθεσης.....	129
Πίνακας 8.10: Αποτελέσματα Post Static Timing.....	129
Πίνακας 8.11: Αναφορά σύνθεσης.....	131
Πίνακας 8.12: Αναφορά MAP.....	131
Πίνακας 8.13: Αναφορά PLace And Route .....	132
Πίνακας 8.14: Αποτελέσματα Post Place And Route Static Timing .....	134
Πίνακας 8.15: Αναφορά σύνθεσης.....	136
Πίνακας 8.16: Αναφορά PLace And Route .....	136
Πίνακας 8.17: Αποτελέσματα χρονισμού PLace And Route .....	136
Πίνακας 8.18: Αποτελέσματα Post Place And Route Static Timing .....	136
Πίνακας 8.19: Αρχείο χρονικών και χωρικών περιορισμών .....	138
Πίνακας Π.1: Λεπτομερής αναφορά σύνθεσης .....	144
Πίνακας Π.2: Σύνθεση της μηχανής πεπερασμένων καταστάσεων .....	145
Πίνακας Π.3: Αποτελέσματα σύνθεσης.....	146
Πίνακας Π.4: Αποτελέσματα MAP.....	146
Πίνακας Π.5: Λεπτομερής αναφορά σύνθεσης .....	147
Πίνακας Π.6: Αναφορά MAP.....	148
Πίνακας Π.7: Αποτελέσματα Place And Route.....	148
Πίνακας Π.8: Αποτελέσματα σύνθεσης.....	149



## Εισαγωγή

### Οργάνωση της διπλωματικής

---

Μια εικόνα περιέχει τεράστιο πλούτο πληροφοριών για οτιδήποτε μας περιβάλλει και αποτελεί ένα σημαντικότερο μέσο μετάδοσης πληροφορίας, στο οποίο στηρίζονται σήμερα σχεδόν όλα τα μέσα μαζικής επικοινωνίας. Με την ανάπτυξη των δικτύων τρίτης και τέταρτης γενιάς αναμένεται να αποτελέσει το σημαντικότερο μέσο επικοινωνίας. Χαρακτηριστική είναι μια κινέζικη παροιμία σύμφωνα με την οποία ‘Μια εικόνα αξίζει όσο χίλιες λέξεις’.

Τα τελευταία χρόνια οι απαιτήσεις για εφαρμογές πολυμέσων αυξήθηκαν, ιδίως μετά τον ερχομό των φορητών υπολογιστών και άλλων ηλεκτρονικών συσκευών, καθώς και της μείωσης του κόστους επικοινωνίας. Σήμερα επιτυγχάνεται συνεχής πρόοδος και βελτίωση στον αναπτυσσόμενο ψηφιακό οπτικό κόσμο. Εκτός από τις υπάρχουσες εφαρμογές σε μηχανήματα fax, σαρωτές (scanners), κάμερες ασφαλείας και φορητές βιντεοκάμερες, νέες αγορές αναδύονται στην οπτική βιομηχανία όπως οι ψηφιακές κάμερες (digital still cameras), τα ηλεκτρονικά παιχνίδια, οι κάμερες για προσωπικούς υπολογιστές, για κινητά τηλέφωνα και PDAs, για βιομετρικές εφαρμογές και συστήματα παροχής οδηγικής βοήθειας.

Η απόλυτη όμως πρόκληση σήμερα στην οπτική τεχνολογία είναι η διατήρηση της ποιότητας της εικόνας στις μικρότερες συσκευές καθώς και η ταυτόχρονη μείωση του κόστους παραγωγής. Τα κριτήρια αυτά ικανοποιούνται σήμερα με τη χρήση νέων μέσων καταγραφής τα οποία άλλαξαν το τρόπο σχεδίασης και λειτουργίας συστημάτων καταγραφής εικόνων και video όπως οι ψηφιακές κάμερες. Όλες αυτές οι μικροκαμωμένες συσκευές καθημερινής χρήσεως που τόσο μεταβάλλουν την καθημερινότητα μας, χρησιμοποιούν σαν μέσο καταγραφής έναν οπτικό αισθητήρα.

Με την πάροδο του χρόνου λοιπόν τα μέσα καταγραφής μιας εικόνας άλλαξαν, εξελίχθηκαν και βελτιώθηκαν. από το κλασικό μέσο καταγραφής τη φωτογραφική μηχανή και το φιλμ, σήμερα φθάσαμε να χρησιμοποιούμε μικρές ψηφιακές κάμερες οι οποίες πλέον ενσωματώνονται σε κάθε είδους συσκευή από τα κινητά τηλέφωνα μέχρι τα παιχνίδια, τις κάμερες ασφαλείας και τα συστήματα οδηγικής βοήθειας. Το μέγεθος όλων αυτών των συσκευών που κατακλύζουν την αγορά και έχουν εισβάλει στην καθημερινή μας ζωή τείνει να μικραίνει.

Ωστόσο η εικόνα που παράγεται από έναν αισθητήρα δεν έχει την καλύτερη δυνατή ποιότητα. Κατά συνέπεια απαιτείται η επεξεργασία της εικόνας και η εφαρμογή αλγορίθμων που ως στόχο έχουν την βελτίωση της ποιότητας της εικόνας. Έτσι έχει δημιουργηθεί ένα ολόκληρο πεδίο έρευνας με μεγάλο ενδιαφέρον γύρω από αυτούς τους αλγορίθμους αφού η ποιότητα της εικόνας εξαρτάται πλέον από την απόδοση αυτών των αλγορίθμων.

Αντικείμενο της παρούσας διπλωματικής είναι η μελέτη των βασικών αλγορίθμων προ-επεξεργασίας εικόνας και η σχεδίαση και υλοποίηση της διεπαφής ενός οπτικού αισθητήρα. Γενικά μερικά από τα πιο κρίσιμα προβλήματα που εντοπίζονται στη σχεδίαση τέτοιων συστημάτων είναι:

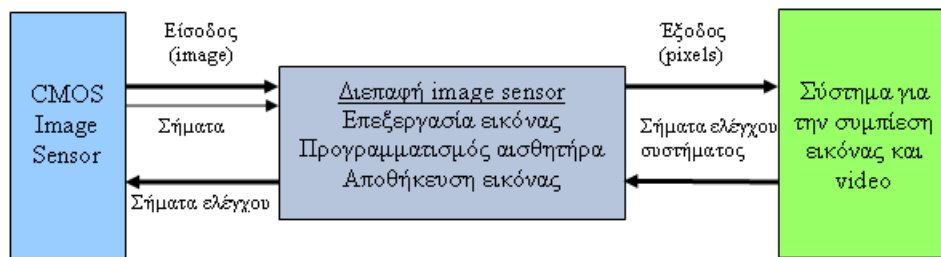
- Από την πλευρά της υλοποίησης, η διατήρηση γρήγορου κλείστρου, η μικρή καθυστέρηση καταγραφής, η χαμηλή κατανάλωση ισχύος, η ταχύτητα εστίασης, η αλληλουχία της επεξεργασίας εικόνας και πάνω από όλα η εκτέλεση αριθμητικής κινητής υποδιαστολής σε ένα ενσωματωμένο σύστημα.
- Από την πλευρά της επιστήμης χρώματος (color science) [13] και της επεξεργασίας εικόνας, η επιλογή φίλτρων, το σχέδιο των φίλτρων CFA, η επιλογή demosaicing τεχνικής, οι αλγόριθμοι αυτόματης έκθεσης (automatic exposure), εξισορρόπησης λευκού (white balance) και αυτόματης εστίασης (auto focus), η βελτίωση της ποιότητας της εικόνας και η επιλογή κατάλληλου χρωματικού χώρου (color space) για την αποθήκευση της εικόνας.

Ο χρήστης είναι αυτός που τελικά καθορίζει τις απαιτήσεις και τα χαρακτηριστικά ενός τέτοιου συστήματος. Η υψηλή ποιότητα εικόνας, η χαμηλή κατανάλωση ισχύος και η γρήγορη επαναπρογραμματιζόμενη (reconfigurable) λειτουργία είναι επιθυμητά χαρακτηριστικά τέτοιων συστημάτων. Ανάλογα με την

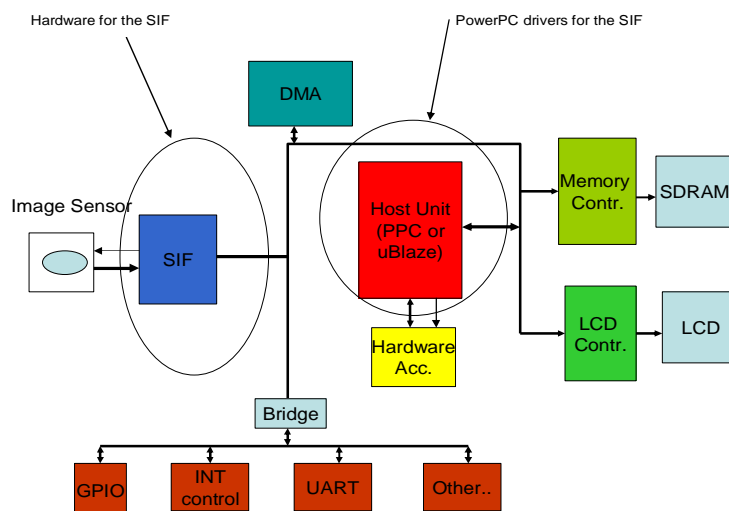
## Εισαγωγή

υλοποίηση του κάθε τμήματος και την επιλογή των αλγορίθμων, η ποιότητα της εικόνας που προκύπτει από ένα τέτοιο σύστημα διαφέρει.

Ένα σύστημα που χρησιμοποιεί οπτικό αισθητήρα για την καταγραφή εικόνας έχει την παρακάτω αρχιτεκτονική. Αρχικά ένα σύστημα διεπαφής μεταξύ του αισθητήρα και του υπόλοιπου συστήματος ελέγχει τον αισθητήρα, δέχεται την εικόνα, την βελτιώνει και την αποθηκεύει ώστε να είναι διαθέσιμη από το υπόλοιπο σύστημα, πιθανή λειτουργία του οποίου είναι η συμπίεση εικόνας και video.



Η διεπαφή του αισθητήρα αποτελείται από δύο τμήματα: ένα τμήμα υλικού (hardware) το οποίο είναι η υλοποίηση της διεπαφής σε μια γλώσσα περιγραφής υλικού και ένα τμήμα λογισμικού (software) το οποίο περιλαμβάνει την υλοποίηση αλγορίθμων επεξεργασίας εικόνας και drivers της διεπαφής και οι οποίοι εκτελούνται σε έναν επεξεργαστή γενικού σκοπού. Η διεπαφή αυτή θα αποτελεί τμήμα ενός System On Chip το οποίο θα μπορεί να χρησιμοποιηθεί για επεξεργασία πολυμέσων, κωδικοποίηση και συμπίεση (compression) εικόνας. Η συνολική αρχιτεκτονική ενός τέτοιου συστήματος απεικονίζεται παρακάτω.



Σχήμα Ε.1: Σύστημα πολυμέσων ενσωματωμένο σε μια FPGA

Στην παρούσα διπλωματική λοιπόν σχεδιάζονται και υλοποιούνται τα επιμέρους κυκλώματα του συστήματος διεπαφής, ακολουθώντας τη ροή πληροφορίας ενσωμάτωσης ενός κυκλώματος σε FPGA. Επιπλέον παρουσιάζεται η αλυσίδα επεξεργασίας εικόνας και μελετούνται και προτείνονται οι σημαντικότεροι αλγόριθμοι προ-επεξεργασίας εικόνας.

### **Οργάνωση της διπλωματικής**

Η διπλωματική αποτελείται από 9 συνολικά κεφάλαια. Αρχικά στο πρώτο κεφάλαιο περιγράφεται η ψηφιακή επεξεργασία εικόνας δίνοντας έμφαση στην αποκαλούμενη αλυσίδα προ-επεξεργασίας, η οποία περιλαμβάνει τους αλγόριθμους εκείνους που βελτιώνουν σημαντικά την ποιότητα μιας εικόνας. Σκοπός του κεφαλαίου είναι να δώσει στον αναγνώστη κάποιες βασικές πληροφορίες ώστε να αποκτήσει κάποιο υπόβαθρο γύρω από τον τομέα προ-επεξεργασίας εικόνας. Έτσι λοιπόν περιγράφεται η δομή μιας εικόνας και οι τρόποι καταγραφής της. Επιπλέον αναφέρονται τα χρωματικά μοντέλα και αναλύονται τα στάδια της αλυσίδας επεξεργασίας εικόνας και τα χαρακτηριστικά των αλγορίθμων κάθε σταδίου.

Στο δεύτερο κεφάλαιο μελετούνται οι αλγόριθμοι προ-επεξεργασίας εικόνας οι οποίοι αποτέλεσαν αντικείμενο ερευνητικής προσπάθειας της διπλωματικής. Περιγράφονται τα βασικά χαρακτηριστικά τους και αναφέρονται ήδη γνωστοί αλγόριθμοι. Επιπλέον περιγράφονται και παρατίθενται τα αποτελέσματα της προσομοίωσης των προτεινόμενων αλγορίθμων που υλοποιήσαμε.

Αντικείμενο του τρίτου κεφαλαίου είναι οι οπτικοί αισθητήρες. Σε αυτό περιγράφονται τα χαρακτηριστικά τους και αναφέρονται οι διαφορές των δύο κυρίαρχων τεχνολογιών κατασκευής αισθητήρων. Επιπλέον περιγράφεται ένας αισθητήρας τελευταίας τεχνολογίας ο οποίος επιλέχθηκε για το σύστημα, επιτρέποντας την πιο ολοκληρωμένη κατανόηση της αρχιτεκτονικής και λειτουργίας ενός παρόμοιου συστήματος.

Στο τέταρτο κεφάλαιο περιγράφεται το πρωτόκολλο I2C το οποίο χρησιμοποιείται για τον προγραμματισμό των καταχωρητών του αισθητήρα. Αναφέρονται τα χαρακτηριστικά του, ο τρόπος λειτουργίας του και προσδιορίζονται τα

πλεονεκτήματά του. Επιπλέον παρατίθενται σχηματικά παραδείγματα της μεταφοράς δεδομένων για την καλύτερη κατανόηση.

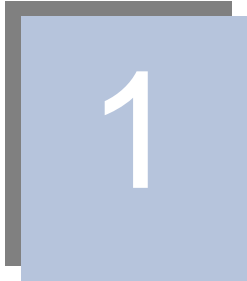
Στο πέμπτο κεφάλαιο περιγράφεται η γενική ροή πληροφορίας σχεδίασης κυκλωμάτων VLSI και η τυπική ροή πληροφορίας για την εγκατάσταση κάποιας σχεδίασης σε FPGA. Επιπλέον αναφέρονται κάποια από τα χαρακτηριστικά της γλώσσας περιγραφής υλικού που χρησιμοποιήσαμε (VHDL) και αναλύονται τα βασικά χαρακτηριστικά ενός FPGA.

Στο έκτο κεφάλαιο περιγράφεται η αρχιτεκτονική του συστήματος διεπαφής που προτείνουμε. Σε κάθε επιμέρους ενότητα περιγράφεται η αρχιτεκτονική κάθε κυκλώματος απαραίτητο για την υλοποίηση του συστήματος στη μορφή μηχανών πεπερασμένων καταστάσεων.

Στο έβδομο κεφάλαιο εφαρμόζονται τα πρώτα στάδια της ροής πληροφορίας για τη σχεδίαση του κυκλώματος. Παρουσιάζεται η περιγραφή του κυκλώματος σε γλώσσα υλικού (VHDL) και τα αποτελέσματα της λειτουργικής προσομοίωσης ενώ αναφέρονται και τα εργαλεία που χρησιμοποιήθηκαν στο στάδιο αυτό.

Στο όγδοο κεφάλαιο συνεχίζουμε με την εφαρμογή των υπολοίπων σταδίων της ροής πληροφορίας στη σχεδίασή μας για την προετοιμασία ενσωμάτωσή της στο FPGA. Για κάθε κύκλωμα ξεχωριστά παρατίθενται τα αποτελέσματα της σύνθεσης, της μετάφρασης, του map και του Place And Route. Επιπλέον απεικονίζεται κάθε κύκλωμα τοποθετημένο στο FPGA.

Τέλος στο ένατο κεφάλαιο περιγράφονται τα αποτελέσματα και η προτεινόμενη μελλοντική εργασία που απαιτείται για την ολοκλήρωση του συνολικού System On Chip, το οποίο θα μπορεί να χρησιμοποιηθεί για επεξεργασία πολυμέσων, κωδικοποίηση και συμπίεση (compression) εικόνας.



## Ψηφιακή επεξεργασία εικόνας

Αντικείμενο του κεφαλαίου είναι η δομή και η επεξεργασία μιας εικόνας. Περιγράφεται ο τρόπος καταγραφής της και αναφέρονται τα χρωματικά μοντέλα που χρησιμοποιούνται και οι τρόποι σάρωσης μιας εικόνας. Επιπλέον αναλύονται τα στάδια της αλυσίδας επεξεργασίας εικόνας και περιγράφονται τα χαρακτηριστικά των αλγορίθμων κάθε σταδίου.



## 1.1 Εισαγωγή

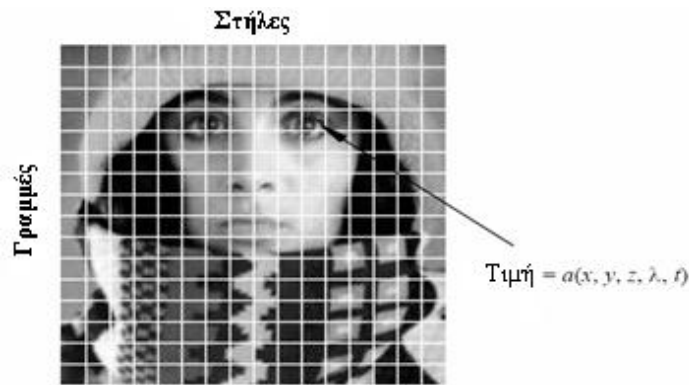
Ο κλάδος που ασχολείται με τον τεράστιο όγκο των οπτικών πληροφοριών και την επεξεργασία των εικόνων ονομάζεται επεξεργασία και ανάλυση εικόνων [2]. Η ψηφιακή επεξεργασία εικόνας είναι ένας ταχύτατα εξελισσόμενος τομέας με αναπτυσσόμενες εφαρμογές στην επιστήμη και την εφαρμοσμένη μηχανική. Ο όρος ψηφιακή επεξεργασία εικόνας γενικά αναφέρεται στην ψηφιακή καταγραφή εικόνας και την επεξεργασία της με έναν ηλεκτρονικό υπολογιστή. Το αντικείμενο της επεξεργασίας μπορεί να είναι η βελτίωση της ποιότητας της εικόνας, το φίλτράρισμα του θορύβου καταγραφής ή μετάδοσης, η συμπίεση του όγκου πληροφορίας, η αποθήκευση εικόνας και η ψηφιακή μετάδοσή της. Η ψηφιακή ανάλυση εικόνας ασχολείται με την περιγραφή και αναγνώριση του περιεχομένου της εικόνας.

Μια ψηφιακή εικόνα είναι ένα πίνακας από πραγματικούς αριθμούς αναπαριστώμενη από ένα πεπερασμένο αριθμό από bits. Η ψηφιοποιημένη εικόνα μπορεί να επεξεργαστεί και να παρουσιαστεί σε μια οθόνη υψηλής ευκρίνειας.

Η ψηφιακή επεξεργασία εικόνας έχει ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβάνοντας τη μετάδοση και αποθήκευση της εικόνας, τη δορυφορική καταγραφή εικόνας, την επεξεργασία ιατρικών, radar και sonar εικόνων, τη ρομποτική και την ανίχνευση και αντίληψη αντικειμένων. Οι εικόνες που ανακτώνται από δορυφόρους είναι χρήσιμες στον εντοπισμό γήινων πόρων, την γεωγραφική χαρτογράφηση, την πρόβλεψη των γεωργικών καλλιεργειών, της αστικής ανάπτυξης και του καιρού. Εφαρμογές μετάδοσης και αποθήκευσης εικόνας παρατηρούνται στην τηλεόραση, τη τηλεδιάσκεψη, τη μετάδοση πανομοιότυπων εικόνων για αυτοματοποίηση γραφείου, στην επικοινωνία μέσω δικτύων υπολογιστών, στα κλειστά κυκλώματα ασφαλείας και στις στρατιωτικές επικοινωνίες. Σε ιατρικές εφαρμογές υπάρχει ενδιαφέρον για την επεξεργασία ακτίνων X, τομογραφιών και άλλων ιατρικών εικόνων που προκύπτουν στην ραδιολογία και τους άλλους ιατρικούς κλάδους. Οι εικόνες αυτές μπορούν να χρησιμοποιηθούν για την ανακάλυψη όγκων και άλλων πιθανών ασθενειών. Οι εικόνες των Radar και των sonar χρησιμοποιούνται για τον εντοπισμό και την αναγνώριση ποικίλων ειδών στόχων ή για την καθοδήγηση αεροπλάνων ή πυραυλικών

συστημάτων. Υπάρχουν πολλές άλλες εφαρμογές από την ρομποτική ως την σύνθεση εικόνων για την δημιουργία κινούμενων σχεδίων ή την σχεδίαση μόδας.

Ο όρος μονόχρωμη εικόνα αναφέρεται ουσιαστικά σε μια δισδιάστατη συνάρτηση έντασης φωτός  $f(x,y)$ , όπου  $x$  και  $y$  δηλώνουν χωρικές συντεταγμένες και η τιμή της  $f$  σε οποιοδήποτε σημείο  $(x,y)$  είναι ανάλογη της φωτεινότητας. Μια ψηφιακή εικόνα είναι μια συνάρτηση  $f(x,y)$  η οποία έχει διακριτοποιηθεί και στις χωρικές συντεταγμένες και στη φωτεινότητα. Μια ψηφιακή εικόνα μπορεί να παρομοιαστεί σαν ένα διάνυσμα του οποίου τα στοιχεία των γραμμών και των στηλών προσδιορίζουν ένα σημείο στην εικόνα και η αντίστοιχη τιμή του στοιχείου του διανύσματος προσδιορίζει το επίπεδο του γκρι σε αυτό το σημείο. Τα στοιχεία ενός τέτοιου ψηφιακού πίνακα καλούνται εικονοστοιχεία ή pixels.



**Σχήμα 1.1:** Πίνακας pixel εικόνας

Το πρώτο βήμα στην επεξεργασία εικόνας είναι η ανάκτηση - καταγραφή μιας εικόνας. Το κλασικό μέσο καταγραφής είναι η φωτογραφική μηχανή και το φιλμ. Σήμερα όμως χρησιμοποιούνται ψηφιακά μέσα καταγραφής όπου απαιτείται ένας οπτικός αισθητήρας. Η πιο βασική απαίτηση για την επεξεργασία των εικόνων είναι η διαθεσιμότητά τους σε ψηφιακή μορφή, δηλαδή σαν πίνακες πεπερασμένου μήκους δυαδικών λέξεων. Για την ψηφιοποίησή της, η εικόνα δειγματοληπτείται σε ένα διακριτό πλέγμα και κάθε δείγμα ή εικονοστίγμα (pixel) ψηφιοποιείται χρησιμοποιώντας ένα πεπερασμένο αριθμό από bits. Μια κοινή μέθοδος δειγματοληψίας εικόνας είναι η σάρωση της εικόνας γραμμή προς γραμμή και η δειγματοληψία κάθε γραμμής και η ψηφιοποίηση του σήματος που παράγεται, από τον αισθητήρα. Η επεξεργασία εικόνας συμπεριλαμβάνει διαδικασίες που συνήθως εκφράζονται σε αλγοριθμική μορφή.

Με εξαίρεση τη καταγραφή και παρουσίαση εικόνας οι περισσότερες λειτουργίες επεξεργασίας μπορούν να υλοποιηθούν σε λογισμικό. Ανάγκη για την χρήση υλικού υπάρχει όταν είναι κρίσιμη η ταχύτητα σε κάποιες εφαρμογές ή για την αντιμετώπιση κάποιων βασικών υπολογιστικών περιορισμών.

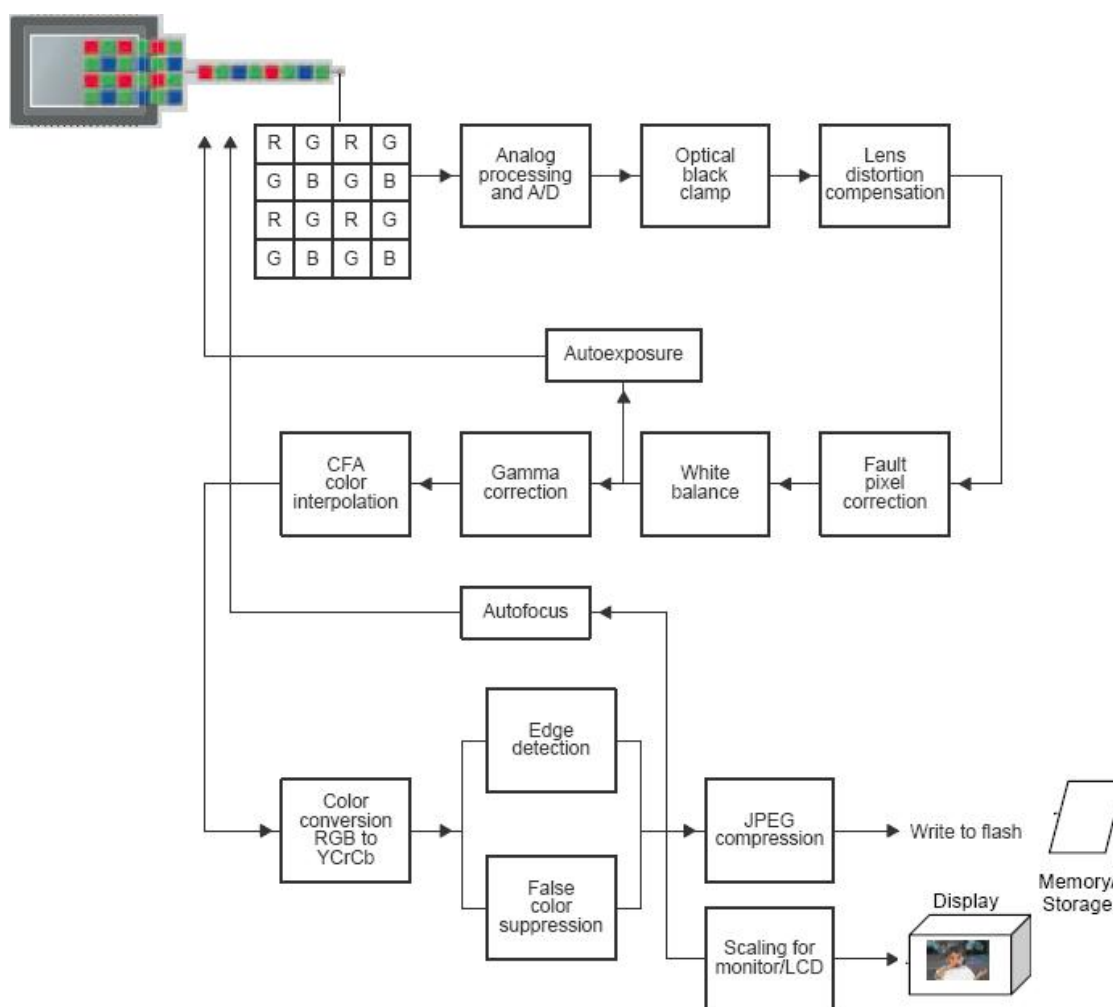
Γενικά το χρώμα του φωτός εξαρτάται από το φασματικό του περιεχόμενο. Κάθε χρώμα μπορεί να δημιουργηθεί συνδυάζοντας 3 βασικά χρώματα. Το πιο συνηθισμένο σύνολο χρωμάτων περιλαμβάνει το κόκκινο, το πράσινο και το μπλε. Αυτό δεν είναι τυχαίο αν αναλογιστεί κανείς ότι το ανθρώπινο μάτι αντιλαμβάνεται τα χρώματα έχοντας υποδοχείς (cones) στον αμφιβληστροειδή χιτώνα που είναι συντονισμένοι στα μήκη κύματος του κόκκινου, του πράσινου και του μπλε.

Η αίσθηση του φωτός μπορεί να περιγραφεί μέσω τριών χαρακτηριστικών: τη φωτεινότητα (luminance), το χρωματικό τόνο (hue), και τον κορεσμό (καθαρότητα χρώματος). Το ανθρώπινο μάτι είναι πιο ευαίσθητο στο φωτισμό, μετά στον χρωματικό τόνο και τελικά στον κορεσμό (saturation). Ένα χρώμα μπορεί να προσδιοριστεί από 3 αριθμούς, είτε αυτούς που αντιστοιχούν στη συμβολή των τριών πρωταρχικών χρωμάτων (π.χ. tristimulus values), είτε μια τιμή φωτεινότητας (luminance) και δύο χρωματικές τιμές (chrominance).

### **1.2 Η αλυσίδα επεξεργασίας εικόνας**

Οι εικόνες που παράγονται από τους αισθητήρες δεν είναι συνήθως βέλτιστες για απεικόνιση ή για περαιτέρω επεξεργασία, πιθανώς εξαιτίας του θορύβου, της χαμηλής αντίθεσης και του θαμπώματος. Υπάρχουν γενικά 3 είδη θορύβου που μπορούν να παρατηρηθούν, ο Fixed pattern noise ο οποίος εμφανίζεται σε μεγάλους χρόνους έκθεσης και κυρίως στους CMOS αισθητήρες, ο τυχαίος θόρυβος με χρωματικές διακυμάνσεις και ο banding θόρυβος που εξαρτάται από το σύστημα της κάμερας. Έτσι λοιπόν απαιτείται η βελτίωση της ποιότητας της εικόνας πριν γίνει μετά επεξεργασία, τελική συμπίεση και απεικόνισή της. Εφαρμόζονται λοιπόν μια σειρά από αλγορίθμους η λεγόμενη αλυσίδα προ-επεξεργασίας εικόνας με σκοπό τη βελτίωση της εικόνας πριν την αποθήκευσή της [9].

Η προ-επεξεργασία αποτελεί ένα σημαντικό πεδίο έρευνας αφού η ποιότητα της εικόνας εξαρτάται από την απόδοση των αλγορίθμων επεξεργασίας εικόνας [8]. Οι αλγόριθμοι εφαρμόζονται είτε μετά την καταγραφή της εικόνας από κάποιο σύστημα που επικοινωνεί με τον αισθητήρα είτε από τον ίδιο αισθητήρα. Τα τελευταία δύο χρόνια εξάλλου άρχισαν να αναπτύσσονται αισθητήρες οι οποίοι ενσωματώνουν επεξεργαστές εικόνας για να εκτελούν τους αλγορίθμους επεξεργασίας εικόνας. Η ενσωμάτωση του συστήματος καταγραφής και της επεξεργασίας στο ίδιο μονολιθικό chip έχει πολλά πλεονεκτήματα συμπεριλαμβανομένου του χαμηλού κόστους κατασκευής, της χαμηλής κατανάλωσης ισχύος, συμπαγή μέγεθος και γρήγορο χρόνο επεξεργασίας. Πολλά από τα block της αλυσίδας μεταβάλλονται ανάλογα με το σύστημα και τον σχεδιαστή.



Σχήμα 1.2: Τυπική αλυσίδα προεπεξεργασίας εικόνας

### 1.3 Καταγραφή εικόνας

Αρχικά λοιπόν μια οπτική συσκευή, δηλαδή ένας οπτικός αισθητήρας καταγράφει την εικόνα, η φωτεινότητα της οποίας μετατρέπεται σε ηλεκτρικό ρεύμα το οποίο μετατρέπεται σε διακριτό δισδιάστατο ψηφιακό σήμα από τον ψηφιοποιητή. Είναι αξιοσημείωτο ότι οι οπτικοί αισθητήρες καταγράφουν μόνο την κλίμακα του γκρι (gray scale), ουσιαστικά συλλαμβάνουν ένα εύρος φωτεινότητας.



Σχήμα 1.3: Κλίμακα του γκρι

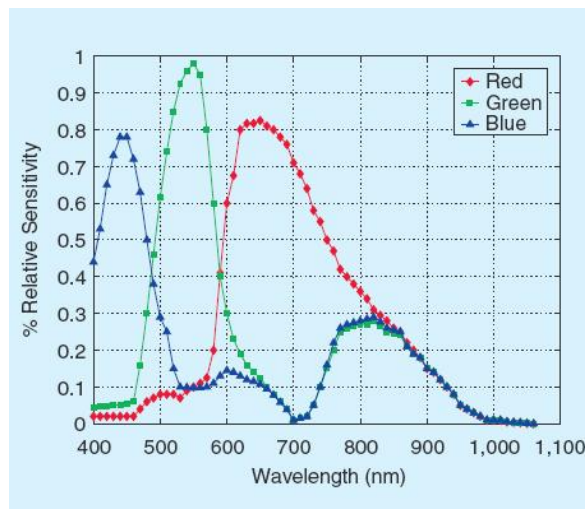
Σε ένα διανυσματικό μοντέλο για οπτικά συστήματα, το ανακλώμενο φάσμα (Reflectance spectrum)  $r(\lambda)$  δειγματοληπτούμενο ομοιόμορφα σε ένα φασματικό εύρος  $[\lambda_{\min}, \lambda_{\max}]$  αλληλεπιδρά με το φωτεινό φάσμα 'illuminant spectrum'  $L(\lambda)$  για τον σχηματισμό μιας απεικόνισης στο χρωματικό χώρο  $RGB_C$  [8] σαν

$$c = N(S^T L M r + n) \quad (1),$$

όπου  $S$  είναι ένα διάνυσμα που προκύπτει από την φασματική ευαισθησία των  $K$  φίλτρων του συστήματος,  $L$  είναι ένα διαγώνιο διάνυσμα με δείγματα από το φάσμα φωτεινότητας στην διαγώνιό του,  $M$  είναι ένα άλλο διαγώνιο διάνυσμα με δείγματα από το φάσμα ευαισθησίας του αισθητήρα (spectrum sensitivity),  $r$  είναι ένα διάνυσμα που αντιστοιχεί στο φάσμα της ανακλώμενης επιφάνειας (surface spectrum reflectance) του αντικειμένου και  $n$  είναι προσθετικός θόρυβος. Το  $N$  αντιστοιχεί στη μη γραμμικότητα του συστήματος. Η εικόνα μπορεί να θεωρηθεί σαν ένας δισδιάστατος πίνακας με διανυσματικές τιμές των pixel σε κάθε θέση. Η εικόνα που σχηματίζεται μοντελοποιείται τότε ως εξής:  $g = B\{Hf\}$  (2), όπου  $B$  είναι ένα χρωματικό φίλτρο (CFA),  $H$  είναι η point spread συνάρτηση που αντιστοιχεί στο οπτικό σύστημα και  $f$  είναι μια αναπαράσταση της εικόνας στην οποία κάθε pixel σχηματίζεται σύμφωνα με την (1).

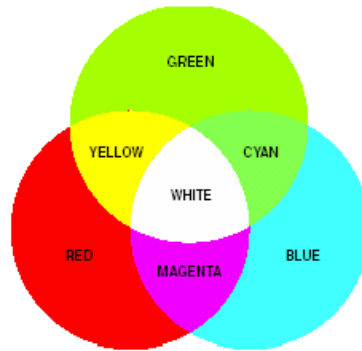
Σύμφωνα με τη τριχρωματική θεωρία ισχύει  $C = \sum_{k=1,2,3} T_k C_k$ , όπου  $T_k$  είναι οι ποσότητες των 3 πρωταρχικών χρωμάτων που απαιτούνται για τη δημιουργία του

χρώματος  $C$  και το  $C_k$ ,  $k = 1, 2, 3$  αναπαριστά τα χρώματα τριών χρωματικών πηγών. Ένας χρωματικός χώρος (color space) [14] είναι μια μαθηματική αναπαράσταση ενός συνόλου χρωμάτων. Τα τρία πιο βασικά χρωματικά μοντέλα είναι το RGB το οποίο χρησιμοποιείται σε γραφικά, το YIQ, YUV, ή YCbCr που χρησιμοποιείται σε συστήματα video και το CMYK το οποίο χρησιμοποιείται στις έγχρωμες εκτυπώσεις. Υπάρχουν συγκεκριμένοι μετασχηματισμοί για τη μετατροπή από το ένα χρωματικό χώρο σε άλλο.



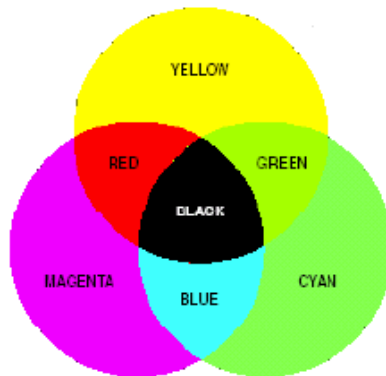
Σχήμα 1.4: Φάσμα ευαισθησίας χρωματικών φίλτρων [8]

Μια ερώτηση που προκύπτει είναι πώς τελικά οι αισθητήρες καταγράφουν χρώματα [12]. Αυτό που συμβαίνει είναι η τοποθέτηση χρωματικών φίλτρων [46] πάνω σε κάθε photosite τα οποία έχουν την ικανότητα διαχωρισμού των χρωμάτων που αντανακλώνται από μια σκηνή. Στους περισσότερους αισθητήρες τα φίλτρα είναι κόκκινου, μπλε και πράσινου χρώματος, τα λεγόμενα RGB (πρωταρχικά χρώματα), ενώ κάποιοι πιο ευαίσθητοι αισθητήρες χρησιμοποιούν φίλτρα cyan, magenta, και κίτρινου (CMYK) χρώματος (συμπληρωματικά χρώματα) [15]. Τα πρωταρχικά φίλτρα επιδεικνύουν άριστη αναπαραγωγή των χρωμάτων αλλά είναι λιγότερα ευαίσθητα ενώ τα συμπληρωματικά είναι περισσότερο ευαίσθητα. Τα τρία πρωταρχικά χρώματα συνιστούν το λεγόμενο additive color system επειδή συνδυάζοντας και τα τρία χρώματα σε ίσες ποσότητες σχηματίζεται το λευκό χρώμα ενώ όταν συνδυάζεται το κόκκινο με το πράσινο σχηματίζεται το κίτρινο χρώμα.



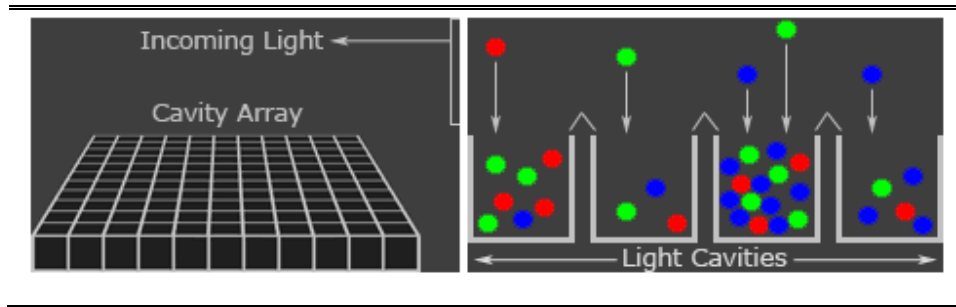
Σχήμα 1.5: Πρωταρχικά χρώματα

Τα συμπληρωματικά χρώματα σχηματίζουν το λεγόμενο subtractive σύστημα όπου αν συνδυαστούν μεταξύ τους σε ίσες ποσότητες σχηματίζεται το μαύρο χρώμα. Χρησιμοποιούνται κυρίως στην εκτύπωση εικόνων και όχι στην απεικόνιση σε οθόνη.

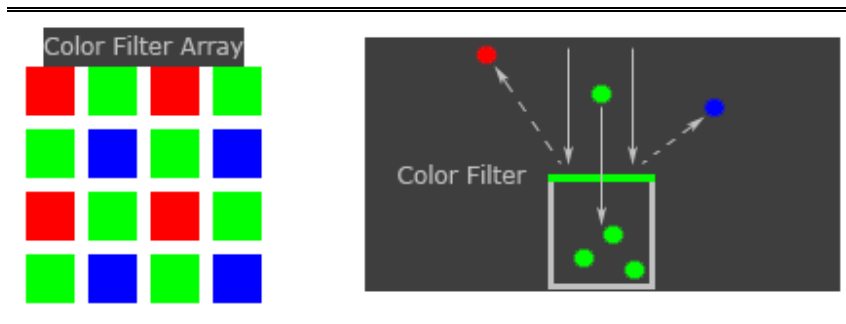


Σχήμα 1.6: Συμπληρωματικά χρώματα

Τα πρωταρχικά φίλτρα τοποθετούνται συνήθως σε μια διάταξη γνωστή ως Bayer pattern [45], [46] η οποία φαίνεται παρακάτω. Δηλαδή μισά από όλα τα pixels είναι πράσινα, το  $\frac{1}{4}$  είναι μπλε και τα υπόλοιπα είναι κόκκινα. Αυτή η συγκεκριμένη διάταξη βασίζεται στην υψηλότερη ευαισθησία που παρουσιάζουν τα ανθρώπινα μάτια στο πράσινο χρώμα. Έτσι από τον αισθητήρα για κάθε pixel παράγεται μια χρωματική συνιστώσα, κόκκινου, πράσινου ή μπλε χρώματος.

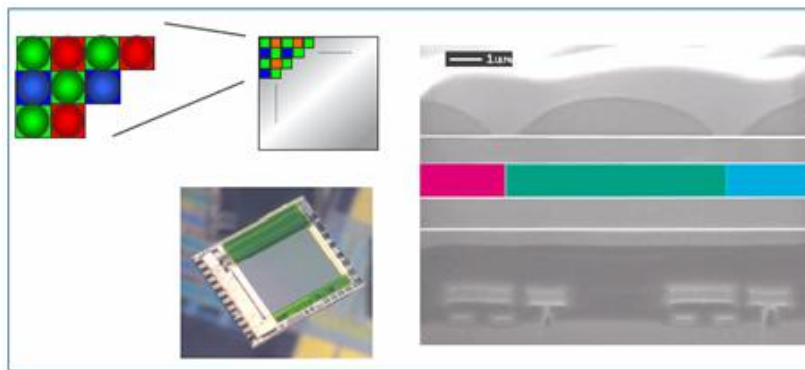


Σχήμα 1.7: Καταγραφή ασπρόμαυρης εικόνας



Σχήμα 1.8: Καταγραφή έγχρωμης εικόνας - Bayer pattern

Συνήθως πάνω από τα φίλτρα χρησιμοποιούνται φακοί για την ενίσχυση και τη βελτίωση σύλληψης της σκηνής.



Σχήμα 1.9: Χρήση φακών - Bayer pattern [12]

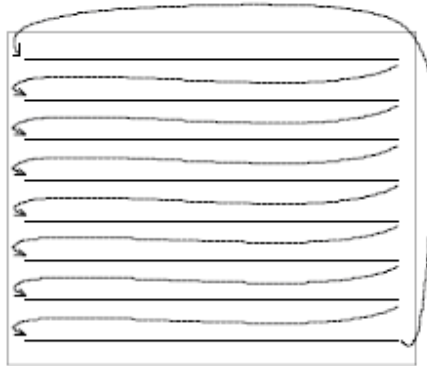
Όταν ο αισθητήρας καταγράφει μια εικόνα πρέπει να τη διαβάσει και να τη μετατρέψει σε ψηφιακή. Το διάβασμα των ηλεκτρικών φορτίων γίνεται γραμμή - γραμμή. Υπάρχουν 2 τρόποι. Σε μια interlaced σάρωση γίνεται ανάγνωση πρώτα των μονών γραμμών και έπειτα των ζυγών. Χρησιμοποιείται συχνά σε video κάμερες ενώ και οι τηλεοπτικές μεταδόσεις είναι αυτού του είδους.





Σχήμα 1.10: Interlaced σάρωση

Σε μια progressive σάρωση γίνεται ανάγνωση των γραμμών η μια με την άλλη. Με αυτό το τρόπο επιτυγχάνεται μεγαλύτερη ανάλυση της εικόνας.



Σχήμα 1.11: Progressive σάρωση

#### 1.4 Στάδια αλυσίδας επεξεργασίας

Αφού λοιπόν καταγραφεί μια εικόνα απαιτείται τα δεδομένα των φίλτρων CFA να υποστούν σημαντική επεξεργασία [8] για να είναι διαθέσιμα για συμπίεση ή αποθήκευση. Αρχικά λοιπόν το ηλεκτρικό σήμα του αισθητήρα οδηγείται σε ένα ψηφιοποιητή στον οποίο το αναλογικό σήμα μετατρέπεται σε ψηφιακό με τη χρήση ενός A/D μετατροπέα. Έτσι η εικόνα μετατρέπεται σε ένα πίνακα με pixel. Στην περίπτωση της έγχρωμης εικόνας αποθηκεύονται τα 3 βασικά χρώματα RGB.

Αρχικό στάδιο στην επεξεργασία εικόνας (σχήμα 2) είναι η ρύθμιση του επιπέδου του μαύρου χρώματος (*Black Clamp*), όπου αφού καθοριστεί το μαύρο στη συνέχεια γίνεται αφαίρεση από κάθε λανθασμένο pixel, ώστε να γίνει σωστή αναπαράσταση του μαύρου. Αυτό γίνεται γιατί στον αισθητήρα για κάθε pixel με μαύρο χρώμα συνεχίζει να ρέει κάποιο ρεύμα και έτσι κάποια pixel έχουν λάθος επίπεδο μαύρου χρώματος.

Κάποιες ατέλειες στους φακούς προκαλούν μη γραμμικότητες στην φωτεινότητα της εικόνας, οι οποίες μειώνουν τη φωτεινότητα ανάμεσα στο κέντρο και στις άκρες της εικόνας. Έτσι το στάδιο επανόρθωσης των ατελειών των φακών (*Lens Distortion Compensation*) ρυθμίζει τη φωτεινότητα κάθε pixel ανάλογα με τη θέση στην οποία βρίσκεται.

Υπάρχει περίπτωση κάποια από τα pixels να είναι κατεστραμμένα οπότε μέσω ενός σταδίου παρεμβολής (*fault pixel interpolation*) γνωστοποιούνται οι τιμές στους στα υπόλοιπα στάδια της αλυσίδας. Συνήθως τα κατεστραμμένα pixel γνωστοποιούνται από τον κατασκευαστή του αισθητήρα, είτε εντοπίζονται καταγράφοντας μια εικόνα χωρίς τη χρήση φακών οπότε τα λανθασμένα pixel φαίνονται σαν λευκά σημεία και η υπόλοιπη εικόνα είναι σκούρα. Εφαρμόζονται επίσης τεχνικές βασισμένες σε φιλτράρισμα διαμέσου (*median-filtering*) για να ανιχνευθούν και να διορθωθούν τα λανθασμένα pixels, γνωστή ως *defective pixel masking* τεχνική.

Η εξισορρόπηση του λευκού (*White Balance*) είναι η διαδικασία της διόρθωσης μιας σκηνής ώστε να δείχνει φυσική, ανεξάρτητα από το φωτισμό που χρησιμοποιείται κατά τη διάρκεια καταγραφής ο οποίος μπορεί να μην είναι κατάλληλος. Ενώ το μάτι κάνει αυτόματα αυτή τη λειτουργία ο αισθητήρας πρέπει να εφαρμόσει αυτή τη λειτουργία ώστε να απεικονιστεί σωστά η εικόνα.[17]. Η εξισορρόπηση επιτυγχάνεται ρυθμίζοντας το σχετικό κέρδος (*gain*) κάθε R, G και B χρωματικού καναλιού. Αρχικά λοιπόν υπολογίζεται η μέση τιμή της φωτεινότητας κάθε χρωματικής συνιστώσας και προσδιορίζεται ένας παράγοντας κλιμάκωσης για κάθε χρωματική συνιστώσα και εφαρμόζεται σε κάθε pixel. Μια άλλη τεχνική υπολογίζει την ενέργεια κάθε χρώματος και την εξισορροπεί. Αυτό το στάδιο υπολογίζει επίσης και ένα γενικό ψηφιακό κέρδος (*digital gain*) που χρησιμοποιείται από τον έλεγχο έκθεσης, AEC. Επιπλέον υπολογίζονται στατιστικά (*pixel statistics*) για όλη την εικόνα ή για μια περιοχή της τα οποία χρησιμοποιούνται για την εκτέλεση εξισορρόπησης του λευκού, αυτόματο έλεγχο της έκθεσης ή για ειδικές διορθώσεις από τον χρήστη.

Η λειτουργία *color interpolation* αποκαλείται *de-mosaicing* και αποτελεί βασικό αλγόριθμο ενίσχυσης της ποιότητας της εικόνας [32]. Ο αισθητήρας για τη σύλληψη εικόνας χρησιμοποιεί ένα χρωματικό φίλτρο για κάθε pixel. Το φίλτρο αυτό (CFA)

επιτρέπει στον αισθητήρα να μετρήσει την τιμή ενός μόνο χρώματος σε κάθε θέση, κόκκινου, πράσινου ή μπλε στην περίπτωση του Bayer format.

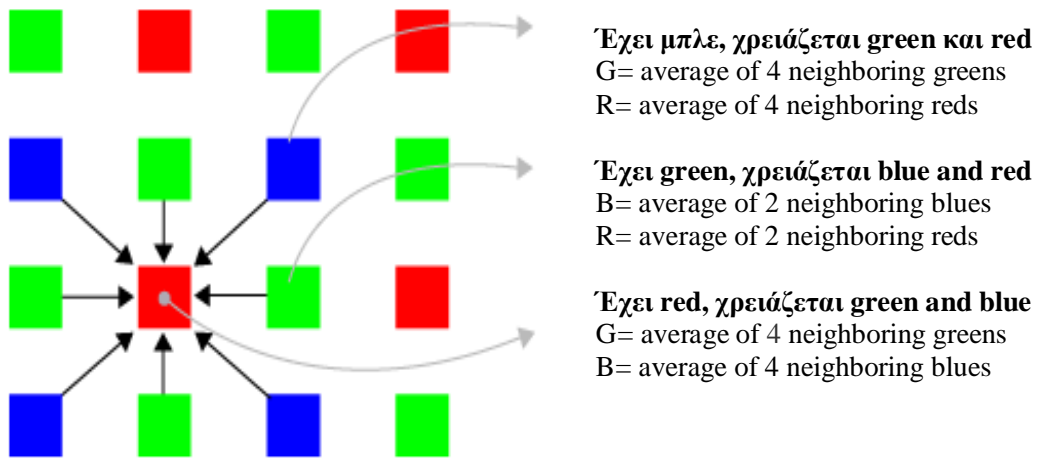


Σχήμα 1.12: Color Interpolation -Bayer format

Η αλυσίδα όμως έχει ανάγκη χρήσης πλήρης ανάλυσης χρώματος για κάθε pixel. Έτσι απαιτείται ένα στάδιο μετατροπής της CFA εικόνας σε πλήρη έγχρωμη εικόνα με 3 χρωματικές συνιστώσες σε κάθε θέση. Ο αλγόριθμος αυτός ονομάζεται *color interpolation*, γιατί πρέπει να προβλέψει τις τιμές των χρωμάτων για να εκτιμήσει τις τιμές των χρωμάτων που λείπουν από κάθε θέση με βάση τα χρώματα των γειτονικών pixel. Υπάρχουν διάφοροι αλγόριθμοι για την υλοποίηση του σταδίου αυτού συμπεριλαμβανομένου των τεχνικών που βασίζονται σε nearest neighbor, linear, cubic και cubic spline .



Σχήμα 1.13: Εικόνα πριν και μετά την εφαρμογή interpolation



Σχήμα 1.14: Τεχνική color interpolation

Όσον αφορά το demosaicking αποτελεί την πιο απαιτητική υπολογιστικά διαδικασία στην αλυσίδα της επεξεργασίας εικόνας. Οι τεχνικές που χρησιμοποιούνται είναι αποτέλεσμα πατεντών [30], [31] και χρησιμοποιούν την πληροφορία neighborhood pixel για να εκτιμήσουν τις τιμές των pixel. Στο [8] γίνεται αναφορά των τεχνικών που χρησιμοποιούνται.

Οι συσκευές απεικόνισης όπως οι CRT οθόνες και οι εκτυπωτές χρησιμοποιούν μια μη γραμμική σχέση συσχέτισης των χρωματικών επιπέδων με την πραγματική ένταση κάθε χρώματος στο pixel. Η *διόρθωση gamma* είναι μια μη γραμμική συνάρτηση που εφαρμόζεται σε κάθε pixel και αντισταθμίζει τις διαφορές ανάμεσα στις εικόνες που παράγονται από τον αισθητήρα και την εικόνα που αναπαρίσταται στην οθόνη.

Συνήθως οι αλγόριθμοι συμπίεσης όπως το JPEG λειτουργεί στον χώρο YCbCr. Έτσι ανάλογα με τον σκοπό χρήσης της εικόνας το στάδιο μετατροπής χρωματικών χώρων (*Color Space Conversion*) μετατρέπει τον χώρο RGB σε YCbCr και αντίστροφα. Η μετατροπή αυτή είναι ένας γραμμικός μετασχηματισμός κάθε Y, Cb και Cr τιμής σαν ένα βεβαρημένο άθροισμα των R, G, B τιμών κάθε pixel.

Η φύση των CFA φίλτρων εισάγει ένα βαθυπερατό φίλτρο το οποίο λειαίνει τις ακμές για την όξυνσή των οποίων χρησιμοποιείται ένας ανιχνευτής ακμών για τον υπολογισμό του μεγέθους τους στο κανάλι Y σε κάθε pixel. Στη συνέχεια το μέγεθος (magnitude) της ακμής κλιμακώνεται και προστίθεται στην αρχική εικόνα (luminance (Y)) για την ενίσχυση των ακμών (sharpness) εικόνας (*Edge Enhancement*). Επειδή η

τεχνική interpolation απαιτεί την χρήση της μέσης τιμής κάθε χρώματος από τα γειτονικά pixels έχει σαν συνέπεια το θάμπωμα της εικόνας. Έτσι χρησιμοποιείται η τεχνική “Aperture correction” η οποία ξεθαμπώνει την εικόνα και επιτυγχάνεται αυξάνοντας το κέρδος των συνιστωσών υψηλών συχνοτήτων με αποτέλεσμα την όξυνση των ακμών και την απομάκρυνση του θορύβου.

Η ενίσχυση των ακμών εφαρμόζεται μόνο στο κανάλι Y της εικόνας. Αυτό έχει σαν αποτέλεσμα την εμφάνιση μη ευθυγραμμίσεων (misalignment) των χρωματικών καναλιών στις ακμές με συνέπεια την εμφάνιση σφαλμάτων σαν ουράνιο τόξο γνωστά ως rainbow artifacts. Η καταπίεση των χρωματικών καναλιών Cb and Cr στις ακμές αφαιρεί αυτά τα σφάλματα.

Είναι επίσης δυνατή η ρύθμιση της εστιακής απόστασης των φακών (Autofocus) ώστε να επιτευχθεί η καλύτερη δυνατή καταγραφή της σκηνής. Οι τεχνικές Auto-Focus ποικίλουν ανάλογα με τον κατασκευαστή κάθε συστήματος [32], [33]. Ο αλγόριθμος αυτός επηρεάζει τη sharpness της εικόνας και ουσιαστικά υπολογίζει μια τιμή της μέγιστης συχνότητας και αλλάζει τη παράμετρο focus μέχρι η μέτρηση να γίνει ελάχιστη.

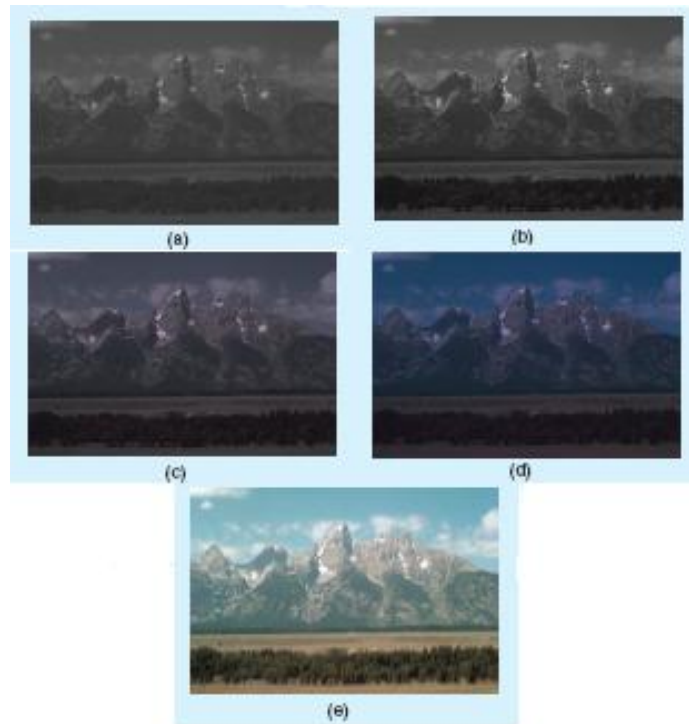
Επειδή ο φωτισμός της σκηνής ποικίλει είναι απαραίτητος ο έλεγχος του χρόνου έκθεσης του αισθητήρα στο φως [16]. Αυτό μπορεί να επιτευχθεί υπολογίζοντας τη μέση τιμή της φωτεινότητας της σκηνής και κατάλληλα να ρυθμίζεται ο χρόνος έκθεσης, Automatic Exposure Control (AEC).

Τέλος εφόσον έχουν εφαρμοστεί τα στάδια της αλυσίδας και έχει προκύψει η βέλτιστη εικόνα είναι δυνατή η συμπίεσή της (Image Compression) εφαρμόζοντας DCT, Jpeg2000 συμπίεση εικόνας ή κάποιο από τα standard συμπίεσης video, MPEG. Όσον αφορά την αποθήκευση της εικόνας είτε χρησιμοποιείται συμπίεση εικόνας μέσω του standard JPEG είτε του νέου βελτιωμένου standard JPEG2000 είτε χρησιμοποιούνται οι νέες τεχνικές υλοποίησης που ακολουθούν οι σύγχρονοι κατασκευαστές υλοποιώντας μεγαλύτερο χώρο για αποθήκευση ή αποθηκεύοντας σε μια ειδική μορφή, επονομαζόμενη TIFF/EP.

Γενικά τα φίλτρα που χρησιμοποιούνται στην βελτίωση και αποκατάσταση μιας εικόνας ανήκουν σε 2 κατηγορίες, στα γραμμικά και μη γραμμικά. Τα γραμμικά φίλτρα μπορούν να σχεδιαστούν και να υλοποιηθούν είτε στο πεδίο της συχνότητας είτε στο πεδίο χωρικών συντεταγμένων. Στην πλειοψηφία τους τα φίλτρα αποκατάστασης είναι

γραμμικά ενώ για τη βελτίωση της ποιότητας της εικόνας χρησιμοποιούνται υλοποιήσεις στο πεδίο των χωρικών συντεταγμένων (γραμμικές ή μη γραμμικές) [2].

Στην επόμενη εικόνα φαίνεται η αρχική εικόνα και η βελτίωση που επιδέχεται ύστερα από την εφαρμογή της αλυσίδας της επεξεργασίας εικόνας.



**Σχήμα 1.15:** Βελτίωση εικόνας με εφαρμογή αλυσίδας επεξεργασίας [8]  
(α) Αρχική εικόνα. (β) Απομάκρυνση θορύβου (black noise). (γ) Μετά το demosaicing.  
(δ) Μετατροπή σε RGB. (ε) Εφαρμογή 3x3 median φίλτρου

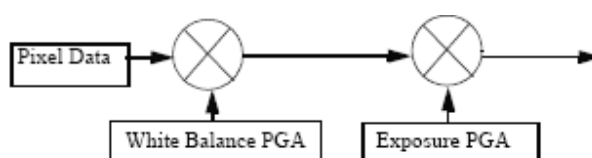


## Αλγόριθμοι προ επεξεργασίας

Στο παρόν κεφάλαιο μελετούνται οι αλγόριθμοι προ-επεξεργασίας εικόνας οι οποίοι αποτέλεσαν αντικείμενο ερευνητικής προσπάθειας της διπλωματικής. Περιγράφονται τα βασικά χαρακτηριστικά τους και αναφέρονται ήδη γνωστοί αλγόριθμοι. Επιπλέον περιγράφονται και παρατίθενται τα αποτελέσματα της προσομοίωσης των αλγορίθμων που υλοποιήσαμε.

## 2.1 Εισαγωγή

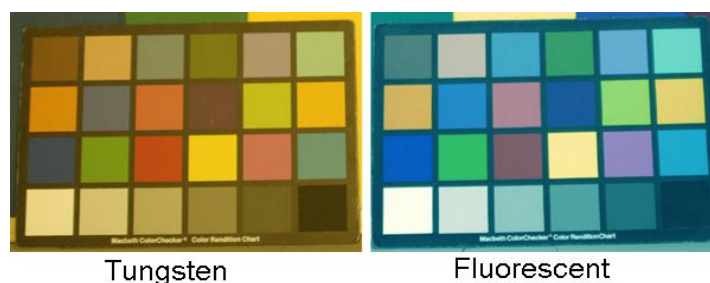
Όπως αναφέρθηκε στο πρώτο κεφάλαιο, αφού καταγραφεί η πραγματική εικόνα, τα ακατέργαστα δεδομένα που λαμβάνονται από τον αισθητήρα χρειάζεται να επεξεργαστούν ώστε να απομακρυνθεί ο θόρυβος και άλλα σφάλματα που επηρεάζουν την ποιότητα της εικόνας, ώστε μέσα από την αλυσίδα επεξεργασίας να δημιουργηθεί μια ακριβής αναπαράσταση της σκηνής. Κάποιοι από τους πιο σημαντικούς αλγόριθμους προ-επεξεργασίας είναι οι αλγόριθμοι εξισορρόπησης του λευκού (white balance) και αυτόματης έκθεσης (auto exposure) οι οποίοι αποτέλεσαν αντικείμενο έρευνας της διπλωματικής.



Σχήμα 2.1: Αλγόριθμοι προ επεξεργασίας εικόνας

## 2.2 Αλγόριθμος εξισορρόπησης του λευκού (White Balance - AWB)

Η ανθρώπινη όραση έχει την ικανότητα εντοπισμού «λευκών» χρωμάτων, δηλαδή αν κάποιος έβλεπε ένα λευκό φύλλο χαρτιού σε φθοριζόμενο (fluorescent – πράσινη απόχρωση) φως ή σε incandescent φως ή ακόμα και σε φυσικό φως θα έλεγε ότι είναι λευκό αν και η ακτινοβολούμενη ενέργεια παράγει διαφορετικούς χρωματισμούς σε διαφορετικούς φωτισμούς. Το φαινόμενο αυτό ονομάζεται χρωματική σταθερότητα (color constancy).



Σχήμα 2.2: Είδη φωτισμού καταγραφής



Ενώ ένα λευκό φως ροζ απόχρωσης που προσκρούει σε έναν τοίχο γίνεται αντιληπτό από τον άνθρωπο σαν λευκό, ωστόσο το λευκό φως που αντανακλάται από ένα κόκκινο τριαντάφυλλο δεν φαίνεται σαν λευκό. Οι κάμερες δεν έχουν την δυνατότητα βελτίωσης μέσω αιώνιας εξέλιξης οπότε και πρέπει να «εκπαιδευτούν» να μπορούν να διακρίνουν το λευκό κάτω από οποιοδήποτε φωτισμό. Όλα αυτά πρέπει να γίνονται αυτόματα χωρίς την εκ των προτέρων γνώση του φωτισμού της σκηνής. Όταν χρησιμοποιείται φλας αυτή η διαδικασία είναι ευκολότερη αφού γενικά γνωρίζουμε το φωτισμό.

Η διαδικασία λοιπόν Auto-White-Balancing (AWB) εκτελεί αυτή τη λειτουργία, αντισταθμίζει δηλαδή αυτόματα το κυρίαρχο χρώμα της σκηνής και διορθώνει μια σκηνή έτσι ώστε να δείχνει φυσική χωρίς να επηρεάζεται από το φωτισμό.

### 2.2.1 Τεχνικές

Το AWB στηρίζεται στην ανάλυση της εικόνας με σκοπό να αντιστοιχήσει το λευκό χρώμα σε ένα λευκό σημείο αναφοράς. Για να το πετύχει αυτό, οι κλασσικές τεχνικές χρησιμοποιούν είτε γενικές μετρήσεις ενέργειας της σκηνής [8], αναλύοντας τη σχετική κατανομή των ποικίλων χρωματικών καναλιών είτε προσπαθούν να προσαρμόσουν το λευκό χρώμα στην συγκεκριμένη συνθήκη φωτός (π.χ. ηλιοβασίλεμα).

Υπάρχουν γενικά δύο τεχνικές που εφαρμόζονται. Η απλή εφαρμόζει καθορισμένη εξισορρόπηση. Αρχικά καταγράφουμε εικόνες σε λευκό ή απαλό γκρι φόντο σε ένα σύνολο φωτισμών, όπως fluorescent, incandescent (tungsten), φως ημέρας και με χρήση φλας. Στη συνέχεια υπολογίζεται η μέση τιμή κάθε χρώματος και η αναλογία μεταξύ κάθε χρώματος και του χρώματος αναφοράς (συνήθως πράσινο), η οποία και αποθηκεύεται. Στη συνέχεια ο χρήστης προσδιορίζει στην αλυσίδα το είδος φωτισμού στο οποίο έγινε η καταγραφή και το υλοποιημένο πρόγραμμα ανακτά τις κατάλληλες αποθηκευμένες αναλογίες. Για παράδειγμα ας υποθέσουμε ότι έχουμε τους μέσους όρους 153 για το κόκκινο, 120 για πράσινο και 100 για μπλε μιας εικόνας που καταγράφη σε φωτισμό incandescent. Αποθηκεύουμε τις αναλογίες κόκκινο/πράσινο-1.28 και μπλε/πράσινο - 0.83.

Ας υποθέσουμε ότι ένα pixel της εικόνας έχει τις τιμές 210 για κόκκινο, 166 για το πράσινο και 132 για το μπλε. Στη συνέχεια διαιρούμε τις κόκκινη τιμή με την

αποθηκευμένη αναλογία κόκκινο/πράσινο και τη τιμή του μπλε με την αναλογία μπλε/πράσινο. Τελικά προκύπτει 164 για το κόκκινο, 166 για το πράσινο και 159 για το μπλε. Η παραπάνω διαδικασία εφαρμόζεται για όλα τα pixels. Η τεχνική αυτή όμως δεν έχει πάντα τα επιθυμητά αποτελέσματα γιατί δεν γνωρίζουμε πάντα το φωτισμό καταγραφής.

Ένας άλλος τρόπος για την εφαρμογή εξισορρόπησης λευκού είναι να υποθέσουμε ότι ένα λευκό σημείο προσδίδει στην κάμερα μέγιστες τιμές στα 3 χρωματικά κανάλια. Με άλλα λόγια αν τα R, G, B, αναπαριστούν τα κόκκινα, τα πράσινα και τα μπλε κανάλια της εικόνας, η white balanced εικόνα έχει σήματα που δίνονται από  $R/R_{MAX}$ ,  $G/G_{MAX}$  και  $B/B_{MAX}$ . Παρολαυτά το μέγιστο των 3 καναλιών αποτελεί φτωχή εκτίμηση του φωτισμού (illuminant).

Έτσι λοιπόν στις κάμερες τελευταίας τεχνολογίας εφαρμόζεται αυτόματη εξισορρόπηση του λευκού [17] η οποία μπορεί να καταλήξει να είναι εξαιρετικά πολύπλοκη διαδικασία. Μια άλλη προσέγγιση της ιδιότητας της χρωματικής σταθερότητας είναι η υπόθεση 'gray world', σύμφωνα με την οποία όλα τα χρώματα στην εικόνα θα έχουν μέση τιμή το γκρι,  $R = G = B$ . Σύμφωνα με αυτό το σχήμα η white balanced εικόνα έχει σήματα που δίνονται από,  $k_r R$ ,  $G$  και  $k_b B$  όπου  $k_r = G_{mean}/R_{mean}$  και  $k_b = G_{mean}/B_{mean}$ . Έτσι όλα τα μη balanced ουδέτερα χρώματα θα αντιστοιχηθούν σε balanced ουδέτερα. Η τεχνική αυτή αποτυγχάνει όταν το μεγαλύτερο μέρος της εικόνας έχει πολλά χρώματα, για παράδειγμα μια εικόνα ενός χρωματιστού λουλουδιού.

Υπάρχει πιθανότητα όμως εάν σε όλη την εικόνα επικρατεί το πράσινο η μέση τιμή της εικόνας να μην τείνει στο γκρι. Έτσι λοιπόν αυξάνουμε την πολυπλοκότητα του αλγορίθμου προσθέτοντας παραμέτρους. Για παράδειγμα κάνουμε υποθέσεις για το φωτισμό που χρησιμοποιήθηκε και στη συνέχεια αποδίδουμε πιθανότητες σε αυτές, τις οποίες χρησιμοποιούμε σαν βάρη για να προσδιορίσουμε τις χρωματικές αναλογίες που πρέπει να ισχύουν.

Γενικά υπάρχουν 2 μεθοδολογίες για να κάνουμε εκτίμηση του φωτισμού [21], [27]. Μια προσέγγιση είναι η χρήση στατιστικών που προκύπτουν από κάθε εικόνα για τη δημιουργία συντελεστών σε ένα πίνακα μετασχηματισμού (η υπόθεση "Gray World" χρησιμοποιεί αυτή της στρατηγική). Μια δεύτερη προσέγγιση η οποία απαιτεί αρκετή μνήμη είναι ο υπολογισμός ενός συνόλου διανυσμάτων για πιθανούς φωτισμούς της

σκηνής και η χρήση των στατιστικών αυτών για την επιλογή του καλύτερου διανύσματος.



**Σχήμα 2.3:** Εφαρμογή αλγορίθμου εξισορρόπησης λευκού

Υπάρχουν γενικά ευρετικές μέθοδοι γύρω από τα προβλήματα που μπορεί να αντιμετωπίζει ο αλγόριθμος, οι οποίες επιτελούν την διαβάθμιση για λιγότερο χρωματιστές εικόνες και χρησιμοποιούνται ανάλογα με το είδος του συστήματος. Στο [18] ο Kehtarnavaz παρουσιάζει μια τεχνική που ονομάζεται scoring για να υλοποιήσει αυτο white balance. Στο [20] ο Bernard παρουσιάζει μια ιστορική ανασκόπηση των αλγορίθμων. Τέλος, τεχνικές που στηρίζονται στην εκτίμηση του φωτισμού αρχίζουν να αναπτύσσονται στην βιομηχανία.

Ωστόσο δεν είναι απαραίτητο η εξισορρόπηση λευκού να γίνεται πάντα αυτόματα αφού μπορεί να μην έχει το επιθυμητό αποτέλεσμα όταν φωτογραφίζουμε με φθορίζον λευκό ή άλλα φθορίζοντα φώτα. Σε αυτές τις περιπτώσεις μερικές ακριβές κάμερες προσφέρουν τη δυνατότητα χρήσης μιας λευκής επιφάνειας ως σημείο αναφοράς ή χρησιμοποιούν προκαθορισμένο ή χειροκίνητο ορισμό της εξισορρόπησης λευκού.

### 2.2.2 Υπόθεση Gray World

Μια από τις υποθέσεις που χρησιμοποιείται σε τεχνικές εξισορρόπησης χρώματος είναι η λεγόμενη Gray World υπόθεση σύμφωνα με την οποία, δοθέντος μιας εικόνας με επαρκείς παραλλαγές χρώματος, η μέση τιμή του κόκκινου, πράσινου και μπλε πρέπει να είναι ίση με μια κοινή γκριζα τιμή. Όλες οι σκηνές στον πραγματικό κόσμο αφομοιώνονται στο γκρι. Αφού οι παραλλαγές των χρωμάτων είναι τυχαίες και ανεξάρτητες σε μια πραγματική σκηνή θα μπορούσε κανείς να υποστηρίξει, ότι δοθέντος πολλών δειγμάτων ο μέσος όρος θα έτεινε να συγκλίνει στη μέση τιμή που είναι το γκρι.

Οι αλγόριθμοι Color balancing χρησιμοποιούν αυτή την υπόθεση ωθώντας τις εικόνες να έχουν μια ομοιόμορφη μέση γκριζα τιμή για τα συστατικά χρώματα R, G, and B. Για παράδειγμα αν μια εικόνα τραβήχτηκε από μια κάμερα κάτω από κίτρινο χρώμα η εικόνα που θα προέκυπτε θα είχε μια κίτρινη απόχρωση σε όλη την εικόνα. Εφαρμόζοντας την υπόθεση στην προκύπτουσα εικόνα είναι δυνατόν να απομακρύνουμε την κίτρινη απόχρωση και να ξανα αποκτήσουμε τα χρώματα της αυθεντικής σκηνής.

Μια απλή μέθοδος της Gray World Assumption θα ήταν η εύρεση των μέσων τιμών R, G, B της εικόνας. Στη συνέχεια υπολογίζουμε αναλογίες των τιμών αυτών με ένα χρώμα αναφοράς, συνήθως το πράσινο. Υπάρχει περίπτωση όμως εάν σε όλη την εικόνα επικρατεί κάποιο χρώμα, η μέση τιμή της εικόνας να μην τείνει στο γκρι. Κάθε χρώμα στη συνέχεια κλιμακώνεται ανάλογα με την απόκλισή του από τη τιμή αυτή. Οι παράγοντες κλιμάκωσης κάθε χρώματος μπορούν να υπολογιστούν διαιρώντας τη γκρι τιμή με την κατάλληλη μέση τιμή κάθε χρώματος.

Στο επόμενο σχήμα απεικονίζεται μια εικόνα που καταγράφηκε σε φωτιζόμενο φως με κυρίαρχη την ενέργεια της κόκκινης συνιστώσας ενώ στο β φαίνεται η εφαρμογή του αλγορίθμου εξισορρόπησης λευκού που στηρίζεται στην υπόθεση Gray World.



Σχήμα 2.4: Εφαρμογή υπόθεσης Gray World

### 2.2.3 Υλοποίηση

Όπως αναφέραμε εάν ο αισθητήρας βρίσκεται σε χώρο με fluorescent ή incandescent φως τότε η εικόνα έχει υπερτονισμένη την πράσινη συνιστώσα ενώ σε εξωτερικό περιβάλλον κυρίαρχη είναι η μπλε συνιστώσα, οπότε πρέπει να εφαρμόσουμε εξισορρόπηση λευκού για να βελτιώσουμε την εικόνα. Το μοντέλο “gray world” ορίζεται ως εξής:

$$M_k = M_b = M_\pi ,$$

όπου  $M_k$  η μέση τιμή της κόκκινης συνιστώσας στο πλαίσιο,  $M_b$  η μέση τιμή της μπλε συνιστώσας και  $M_p$  η μέση τιμή της πράσινης συνιστώσας στο πλαίσιο. Τονίζουμε ότι πρόκειται για μέση τιμή στο πλαίσιο και όχι για κάθε pixel. Για να εφαρμόσουμε το μοντέλο Gray World θα πρέπει να αυξήσουμε ή να μειώσουμε τη τιμή κάθε pixel. Αν για παράδειγμα υποθέσουμε ότι το χρώμα αναφοράς είναι το κόκκινο τότε θα πρέπει να μεταβάλλουμε τις τιμές της πράσινης και μπλε συνιστώσας κάθε pixel του πλαισίου. Αυτό μπορεί να γίνει χρησιμοποιώντας τα ατομικά κέρδη, digital gains, κάθε χρώματος.

Σημειώνουμε ότι η μεταβολή της τιμής του χρόνου έκθεσης, exposure time, προκαλεί την έναρξη του αλγορίθμου εξισορρόπησης λευκού. Δηλαδή εάν μεταβληθεί η φωτεινότητα του περιβάλλοντος του αισθητήρα, θα μεταβληθεί ο χρόνος έκθεσης και θα πρέπει να ξανά εφαρμόσουμε εξισορρόπηση λευκού αφού θα έχουν μεταβληθεί οι τιμές των χρωμάτων.

Υλοποιήθηκε λοιπόν ο αλγόριθμος εφαρμογής του αλγορίθμου εξισορρόπησης λευκού με τη χρήση του εργαλείου Matlab. Σε αυτό αρχικά διαβάζουμε μια εικόνα που δίνουμε σαν είσοδο και επιστρέφεται ένας τρισδιάστατος πίνακας με τις τιμές κάθε χρώματος σε κάθε διάσταση, των pixel της εικόνας. Στη συνέχεια αφού υπολογίσουμε τη μέση τιμή του κάθε χρώματος υπολογίζουμε μια συνολική μέση τιμή, τη γκρίζα τιμή. Στη συνέχεια υπολογίζουμε ένα παράγοντα κλιμάκωσης κάθε χρώματος, διαιρώντας τη γκρίζα τιμή με τη μέση τιμή κάθε χρώματος. Στη συνέχεια, εφαρμόζουμε την κλιμάκωση σε κάθε χρώμα.

Τα αποτελέσματα από την προσομοίωση του αλγορίθμου με χρήση εικόνων που κατεγράφησαν σε περιβάλλον με fluorescent ή incandescent φως φαίνονται παρακάτω. Στην εικόνα 5γ έγινε κλιμάκωση χρησιμοποιώντας σαν αναφορά το κόκκινο χρώμα, παρατηρώντας ότι η εικόνα διατηρεί μια κόκκινη απόχρωση.



(α): Αρχική εικόνα



(β): Εφαρμογή white balance



(γ): κόκκινο χρώμα σαν σημείο αναφοράς

**Σχήμα 2.5:** Εφαρμογή white balance

Με την εφαρμογή του αλγορίθμου οι τιμές κάθε χρώματος κάθε pixel μεταβάλλονται. Για παράδειγμα το pixel (184, 51) της αρχικής εικόνας έχει τις εξής τιμές [225 211 175]. Το ίδιο pixel με εφαρμογή της εξισορρόπησης λευκού έχει τις εξής τιμές [209 211 208] παρατηρώντας ότι οι τιμές τείνουν να ισοροπήσουν. Τέλος η εικόνα με εφαρμογή της εξισορρόπησης του λευκού χρησιμοποιώντας το κόκκινο σαν αναφορά έχει τιμές [225 211 208] όπου το κόκκινο είναι όπως περιμέναμε τονισμένο.



**Σχήμα 2.6:** (α) Αρχική εικόνα, (β) Εφαρμογή εξισορρόπησης λευκού

Εφαρμόζουμε τον αλγόριθμο στην δημοφιλή εικόνα Lena. Στην εικόνα (γ) διατηρούμε την κόκκινη συνιστώσα αφού κάνουμε την υπόθεση ότι η εικόνα καταγράφηκε σε κόκκινο φωτισμό στην οποία και αποδίδουμε αυξημένη πιθανότητα παρατηρώντας και την εικόνα. Παρατηρούμε στη περίπτωση αυτή πόσο βελτιώθηκε η εικόνα με εισαγωγή στον αλγόριθμο τεχνικής υπόθεσης και απόδοσης πιθανότητας που αναφέραμε παραπάνω.



(α): Αρχική εικόνα



(β): Εφαρμογή εξισορρόπησης λευκού



(γ): White balance με υπόθεση φωτισμού

**Σχήμα 2.7:** Εφαρμογή εξισορρόπησης λευκού στην δημοφιλή εικόνα Lena

Στη συνέχεια φαίνεται μια πολύχρωμη εικόνα που πιθανότατα καταγράφηκε σε εξωτερικό περιβάλλον με πολύ ήλιο. Παρατηρούμε ότι τα χρώματα σχεδόν αλλάζουν. Είναι η περίπτωση που γενικά οι αλγόριθμοι εξισορρόπησης λευκού μπορεί να αντιμετωπίσουν προβλήματα.



(α)



(β)

**Σχήμα 2.8:** (α) Αρχική εικόνα, (β) Εφαρμογή white balance

Στην επόμενη εικόνα το pixel (176, 166) της αρχικής εικόνας έχει τις εξής τιμές [212 61 68]. Το ίδιο pixel με εφαρμογή της εξισορρόπησης λευκού έχει τις εξής τιμές [160 76 78] παρατηρώντας ότι η τιμή του κόκκινου γίνεται λιγότερο έντονη ενώ οι υπόλοιπες αυξάνονται ώστε να επιτευχθεί εξισορρόπηση.



(α)



(β)

**Σχήμα 2.9:** (α) Αρχική εικόνα, (β) Εφαρμογή εξισορρόπησης λευκού



## 2.3 Αλγόριθμος αυτόματης έκθεσης (Auto Exposure - AE)

Η παράμετρος αυτόματης έκθεσης προσδιορίζει τη ποσότητα του φωτός που φθάνει στον αισθητήρα, καθορίζοντας πόσο φωτεινή ή σκοτεινή θα είναι η εικόνα. Όταν ανοίγει το κλείστρο, το φως χτυπά τον αισθητήρα εικόνας [24]. Αν το χτυπά πολύ φως η φωτογραφία θα είναι πολύ εκτεθειμένη στο φως και θα ξεθωριάσει. Λίγο φως παράγει μια σκοτεινή φωτογραφία με έλλειψη λεπτομέρειας κυρίως στις σκιασμένες περιοχές.

Είναι κοινή πρακτική να δίνεται έμφαση στο κάτω μέρος της εικόνας για την μείωση της πιθανότητας να επηρεάσει ο φωτεινός ουρανός τη φωτεινότητα της σκηνής. Άλλοι δίνουν έμφαση στο κέντρο της εικόνας βασιζόμενοι στην υπόθεση ότι το κυρίως αντικείμενο της εικόνας βρίσκεται εκεί. (*center-weighted system*).

Ο έλεγχος της έκθεσης συνήθως απαιτεί χαρακτηρισμό της φωτεινότητας της σκηνής. Γενικά ανάλογα με την μετρούμενη ενέργεια του αισθητήρα, το σύστημα ελέγχου αλλάζει το άνοιγμα του φωτοφράκτη ή/και την ταχύτητα του κλείστρου. Ο έλεγχος της έκθεσης μπορεί να βασίζεται στη πραγματική φωτεινότητα της RGB εικόνας ή μόνο στο πράσινο κανάλι που είναι μια καλή εκτίμηση της luminance συνιστώσας.

### 2.3.1 Τεχνικές

Γενικά υπάρχει λίγη ακαδημαϊκή βιβλιογραφία σε αλγορίθμους αυτόματης έκθεσης, ωστόσο υπάρχουν αρκετές πατέντες [16], [8], [26]. Οι περισσότεροι αλγόριθμοι αυτόματης έκθεσης λειτουργούν ως εξής. Χρησιμοποιείται μια τιμή έκθεσης, exposure value (EV) για να προσδιοριστεί η σχέση ενός αριθμού,  $F$ , και της διάρκειας της έκθεσης,  $T$  (σχέση 1).

$$EV = \log_2\left(\frac{F^2}{T}\right) = 2\log_2(F) - \log_2(T) \quad (1)$$

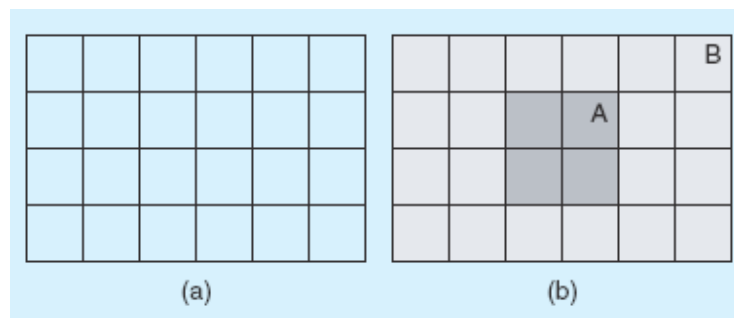
Αρχικά γίνεται καταγραφή μια εικόνας με προκαθορισμένη τιμή έκθεσης ( $EV_{pre}$ ) και μετατρέπονται οι RGB τιμές σε φωτεινότητα,  $B$ . Έπειτα παράγεται ένας αριθμός  $B_{pre}$  από τη φωτεινότητα της εικόνας. Βασιζόμενοι στην υπόθεση γραμμικότητας και τη σχέση 1 η βέλτιστη τιμή έκθεσης  $EV_{opt}$  πρέπει να είναι τέτοια ώστε η εικόνα που θα

καταγραφεί σε αυτό το  $EV_{opt}$  θα μας δώσει έναν αριθμό κοντά σε μια προκαθορισμένη τιμή φωτεινότητας  $B_{opt}$ .

$$EV_{opt} = EV_{pre} + \log 2(B_{pre}) - \log 2(B_{opt}) \quad (2)$$

Για τον καθορισμό της τιμής  $B_{pre}$  υπάρχουν διάφορες τεχνικές οι οποίες υποθέτουν ότι η τιμή  $B_{pre}$  είναι η μέση φωτεινότητα σε όλη την εικόνα ή η τιμή  $B_{pre}$  είναι η βεβαρημένη μέση τιμή του κέντρου της εικόνας ή η τιμή  $B_{pre}$  είναι η μέση τιμή του πράσινου καναλιού.

Σύμφωνα με έναν προτεινόμενο αλγόριθμο [8], [29], η εικόνα χωρίζεται σε τμήματα όπως φαίνεται στην εικόνα 10. Η μέση τιμή της luminance υπολογίζεται για κάθε τμήμα. Σε μια τυπική εικόνα η μέση τιμή συγκρίνεται με μια τιμή αναφοράς και ο χρόνος έκθεσης ελέγχεται για την διατήρηση σταθερής luminance τιμής. Οι σκοτεινές ή φωτεινές εικόνες διακρίνονται μετρώντας την διαφορά με τη μέση τιμή φωτεινότητας του κάθε block (σχήμα 10β). Αν η εικόνα είναι πολύ φωτεινή οι μέσες τιμές της ενέργειας στην περιοχή A θα είναι πολύ υψηλότερη από την ενέργεια της περιοχής B. Αν είναι σκοτεινή η ενέργεια στην περιοχή B θα είναι υψηλότερη. Η έκθεση ελέγχεται για την διατήρηση της διαφοράς της μέσης τιμής των 2 περιοχών, μια εκτίμηση του object contrast.



**Σχήμα 2.10:** Αλγόριθμος - εικόνα χωρισμένη σε 2 τμήματα

Σύμφωνα με μια άλλη τεχνική [16] η οποία δίνει έμφαση στη φωτεινότητα του αντικειμένου ανάλογα με το βαθμό του φόντου, η εικόνα χωρίζεται σε 5 περιοχές. Στις περισσότερες περιπτώσεις η περιοχή του φόντου βρίσκεται στο πάνω τμήμα ενώ το αντικείμενο βρίσκεται στο κέντρο της εικόνας. Ο φωτισμός σε κάθε περιοχή κλιμακώνεται διαφορετικά ανάλογα με το βαθμό του φόντου της περιοχής. Η διαφορά,

luminance difference  $D_b$  για την ανίχνευση του φόντου υπολογίζεται σαν

$$D_b = [R_0 + \max(R_2, R_3)] - (R_1 + R_4),$$

όπου το  $R_i$  δηλώνει την μέση τιμή luminance της περιοχής  $i$  και το  $\max(R_2, R_3)$  χρησιμοποιείται για να καθοριστεί εάν η πηγή φωτός είναι στην αριστερή ή στη δεξιά πλευρά της σκηνής. Για να μετασχηματίσουμε το βαθμό του φόντου στο ομαλοποιημένο διάστημα  $[0,1]$  χρησιμοποιείται μια συνάρτηση μεταφοράς.

Ο φωτισμός των κυρίως περιοχών, 1 και 4 είναι πιο έντονος χρησιμοποιώντας το ομαλοποιημένο βαθμό φόντου,  $N_b$ , από τις περιοχές φόντου. Το συνολικό luminance προκύπτει προσθέτοντας τα βεβαρημένα luminances κάθε περιοχής.

Οι επόμενες εικόνες δείχνουν διαφορετικά πιθανά επίπεδα έκθεσης μιας εικόνας.



**Σχήμα 2.11:** Επίπεδα έκθεσης

(α) υπο εκθετημένη, (β) υπέρ εκθετημένη εικόνα, (γ) καλή έκθεση

### 2.3.2 Υλοποίηση

Ο στόχος για αυτόματη έκθεση (auto exposure) είναι να παραμείνει η μέση τιμή του χρώματος που χρησιμοποιούμε σαν αναφορά μέσα σε κάποια όρια. Για παράδειγμα θα επιθυμούσαμε η μέση τιμή του να διατηρείτε στο εύρος  $[150\ 180]$  για εικόνα 8-bit. Εάν αυτό δεν συμβαίνει για κάποιο πλαίσιο τότε θα πρέπει να μεταβάλλουμε τον χρόνο έκθεσης και να ξανε εφαρμόσουμε εξισορρόπηση λευκού στην εικόνα. Βασικός μας λοιπόν στόχος είναι να θέσουμε τη μέση τιμή του κόκκινου (meanRed) σε κάποια όρια. Ο αλγόριθμος αυτόματης έκθεσης υλοποιήθηκε με τη χρήση του εργαλείου Matlab και φαίνεται στην παρακάτω μορφή ψευδογλώσσας. Στον επόμενο πίνακα φαίνεται το γενικό πρόγραμμα που ελέγχει αν θα πρέπει να γίνει αυτόματη έκθεση, αλλιώς εκτελείται εξισορρόπηση λευκού .

```

Εάν (|meanRed – κατώφλι| > εύρος) {

    /*Εάν η μέση τιμή είναι εκτός ορίων εκτελούμε Auto exposure */
    Άλλαξε Exposure time;
    ColorFlag = 1;
    /* εκτελούμε White Balance μετά,, εκτός Εάν η μέση τιμή είναι και πάλι εκτός ορίων
    */
}

Αλλιώς Εάν (ColorFlag == 1) {

    /* η μέση τιμή ήταν εκτός ορίων πριν αλλά όχι σε αυτό το πλαίσιο. Εκτέλεσε WB */
    Υπολόγισε νέες τιμές WB ;
    ColorFlag = 0;
}
Αλλιώς {
    επέστρεψε (return) /*όχι αλλαγές για το επόμενο πλαίσιο */
}
    
```

**Πίνακας 2.1:** Έλεγχος εφαρμογής αλγορίθμου αυτόματης έκθεσης

Το εύρος τιμών στο οποίο επιθυμούμε να βρίσκεται η μέση τιμή του χρώματος αναφοράς είναι το διάστημα [150 180] για 8-bit αναπαράσταση κάθε χρώματος. Επιπλέον το κατώφλι τίθεται και αυτό από τον χρήστη του αισθητήρα σε μια μέση τιμή, συνήθως 150. Ο αλγόριθμος εύρεσης νέας τιμής χρόνου έκθεσης μπορεί να είναι απλός ή εξαιρετικά πολύπλοκος. Ο αλγόριθμος εύρεσης της νέας τιμής χρόνου έκθεσης που προτείνουμε παρατίθεται στο επόμενο πίνακα.

```

Diff = meanRed – κατώφλι;      /* meanRed = μέση τιμή κόκκινης απόχρωσης */

Εάν (Diff >= 0) {      /* πολύ λαμπερή σκηνή, μείωση του exposure time */

    Εάν (ExposureTime <= MinExposureTime) {
        /* ο χρόνος είναι ήδη αρκετά μικρός. Δεν αλλάζει */
        ExposureTime = ExposureTime;
        return;
    }
    Αλλιώς {

        Εάν (Exposure Time < MinExposureTime) {
            /* ο χρόνος είναι αρκετά μικρός. Τίθεται στη μικρότερη δυνατή τιμή.*/
            Exposure Time = MinExposureTime;
        }
        Αλλιώς{
            /*Υπολογίζεται το Δtime. Μειώνεται το exposure time.*/
            Δtime = (Diff * MaxExposureTime) / (MaximumPixelValue);
            Exposure Time = Exposure Time – Δtime;
        }
    }
}
Αλλιώς {      /* Diff < 0, πολύ σκοτεινή σκηνή. Αύξηση του exposure time */

    Εάν (ExposureTime >= MaxExposureTime) {
        /* ο χρόνος είναι ήδη αρκετά μεγάλος. Δεν αλλάζει */
        ExposureTime = ExposureTime;
        return;
    }
    Αλλιώς {

        Εάν (Exposure Time > MaxExposureTime) {
            /* ο χρόνος είναι αρκετά μεγάλος. Τίθεται στην μέγιστη δυνατή τιμή.*/
            Exposure Time = MaxExposureTime;
        }
        Αλλιώς{
            /*Υπολογίζεται το Δtime. Αυξάνεται το exposure time.*/
            Δtime = (Diff * MaxExposureTime) / (MaximumPixelValue);
            Exposure Time = Exposure Time + Δtime;
        }
    }
}
}

```

**Πίνακας 2.2:** Εύρεση νέου χρόνου έκθεσης σε ψευδογλώσσα

Οι τιμές `MaxExposureTime` και `MinExposureTime` εξαρτώνται από τον αισθητήρα ή τον χρήστη και συνήθως ορίζονται στις τιμές 30 ms και 1ms αντίστοιχα. Επίσης η τιμή `MaximumPixelValue` είναι ίση με 255 ή 1023 για 8 και 10-bit pixels αναπαράσταση, αντίστοιχα.

Γενικά εάν φθάσουμε σε μια μέγιστη τιμή για τον χρόνο έκθεσης τότε μπορούμε να αυξήσουμε μια άλλη παράμετρο, το κέρδος (digital gain) της εικόνας, για να γίνει πιο φωτεινή. Υπάρχει όμως κάποιο όριο σε αυτή τη διαδικασία επειδή υψηλό κέρδος δημιουργεί θόρυβο στο χρωματικό κανάλι και πολλά artifacts στην εικόνα.

Αποτέλεσμα των προγραμμάτων που υλοποιήσαμε είναι η νέα τιμή για τον χρόνο έκθεσης και η εξισορροπημένη εικόνα εφόσον έχει αλλάξει ο χρόνος έκθεσης για το πλαίσιο.

A blue square with a white number '3' inside, positioned on the left side of the page.

## Οπτικοί Αισθητήρες

Στο παρόν κεφάλαιο μελετούνται οι οπτικοί αισθητήρες. Περιγράφονται τα χαρακτηριστικά τους και αναφέρονται οι διαφορές των δύο κυρίαρχων τεχνολογιών κατασκευής αισθητήρων. Επιπλέον περιγράφεται ένας αισθητήρας τελευταίας τεχνολογίας ο οποίος επιλέχθηκε για το σύστημα, επιτρέποντας την πιο ολοκληρωμένη κατανόηση της αρχιτεκτονικής και λειτουργίας ενός τέτοιου συστήματος.

### 3.1 Εισαγωγή

Η πιο βασική απαίτηση για την επεξεργασία μιας εικόνας είναι η διαθεσιμότητά της σε ψηφιακή μορφή, δηλαδή σε πίνακα πεπερασμένου μεγέθους με δυαδικά στοιχεία. Για την ψηφιοποίησή της, η εικόνα δειγματοληπτείται σε ένα διακριτό πλέγμα και κάθε pixel ή δείγμα αναπαρίσταται χρησιμοποιώντας ένα πεπερασμένο αριθμό από bits. Μια κοινή μέθοδος για τη δειγματοληψία της εικόνας είναι η σάρωσή της, γραμμή προς γραμμή και έπειτα η δειγματοληψία της κάθε γραμμής. Κάποιοι από τις πιο συνηθισμένες συσκευές σάρωσης είναι οι οπτικοί αισθητήρες.

Σήμερα χρησιμοποιούνται οπτικοί αισθητήρες σε ολοένα και περισσότερες συσκευές πολλές, από τις οποίες τείνουν να γίνουν αναπόσπαστο τμήμα της καθημερινής μας ζωής. Ψηφιακές κάμερες, κινητά τηλέφωνα και τηλέφωνα video κλήσης είναι μερικές από τις συσκευές όπου χρησιμοποιούνται οι οπτικοί αισθητήρες.



Σχήμα 3.1: Εφαρμογές οπτικών αισθητήρων

Για την κατασκευή αισθητήρων εικόνας υπάρχουν δύο τεχνολογίες:

- η CCD (Charge Coupled Device) τεχνολογία και
- η CMOS (Complementary Metal Oxide Semiconductor) τεχνολογία.



Μέχρι τα μέσα του 1990, η CCD ήταν η κυρίαρχη τεχνολογία στον κόσμο της εικόνας, ενώ παραδοσιακά ενσωματωμένα κυκλώματα κατασκευάζονταν με τη CMOS τεχνολογία. Από τότε όμως άρχισε να διαμορφώνεται μεγάλο ενδιαφέρον στην ανάπτυξη των οπτικών αισθητήρων CMOS εξαιτίας των πλεονεκτημάτων που παρουσίαζαν [46]. Η πρώτη συσκευή CCD αναφέρθηκε στα Bell Labs το 1970. Υιοθετήθηκε σε βάρος άλλων solid-state οπτικών αισθητήρων, εξαιτίας του μειωμένου fixed pattern noise (FPN) και του μικρότερου μεγέθους pixel. Μετά από 30 χρόνια από την υιοθέτησή τους οι CCD οπτικοί αισθητήρες γνώρισαν μεγάλη ανάπτυξη και βελτίωση επιτυγχάνοντας υψηλή απόδοση με χαμηλό θόρυβο (readout noise), και άριστη ανταπόκριση (responsivity).

Σύντομα όμως φάνηκαν και οι λειτουργικοί περιορισμοί των CCDs. Η διαδικασία κατασκευής CCD δεν επιτρέπει την ενσωμάτωση κυκλωμάτων on-chip όπως επεξεργαστές σήματος και μετατροπείς από αναλογικό σε ψηφιακό σήμα οι οποίοι είναι απαραίτητοι για την ψηφιοποίηση της εικόνας εφόσον ενσωματώνονται στον ψηφιοποιητή εικόνας. Σαν αποτέλεσμα μια κάμερα βασισμένη σε CCD δεν απαιτεί μόνο ένα chip οπτικού αισθητήρα, αλλά ένα σύνολο από chips, τα οποία αυξάνουν την κατανάλωση ισχύος και δεν επιτρέπουν μικρά μεγέθη συσκευών. Επιπροσθέτως, η λειτουργία των CCD δεν επιτρέπει τη λεγόμενη “window of interest” ανάγνωση. Μέχρι πρόσφατα λοιπόν, οι charge-coupled devices (CCDs) ήταν οι μόνοι οπτικοί αισθητήρες που χρησιμοποιούνταν σε ψηφιακές κάμερες. Αναπτύχθηκαν και βελτιώθηκαν μέσω της χρήσης τους σε αστρονομικά τηλεσκόπια, scanners και βιντεοκάμερες.

Τα τελευταία όμως χρόνια ανέτειλε μια νέα πρόκληση στον ορίζοντα, οι οπτικοί αισθητήρες CMOS που υπόσχονται ότι θα γίνουν οι αισθητήρες που θα χρησιμοποιούνται στο μεγαλύτερο τμήμα της αγοράς το οποίο ήδη έχει αρχίσει να διαφαίνεται. Κι όντως η έρευνα τα τελευταία χρόνια είχε σαν αποτέλεσμα σημαντικές προόδους στον τομέα των αισθητήρων CMOS, προσφέροντας απόδοση σαν των CCDs, συνδυάζοντας υψηλή λειτουργικότητα και χαμηλή κατανάλωση ισχύος.

Πολλοί είναι οι παράγοντες που συνετέλεσαν στην ανάπτυξη των CMOS αισθητήρων. Κύριος παράγοντας είναι η πρόσφατη απαίτηση των καταναλωτών για φορητά, χαμηλής κατανάλωσης και μικρού μεγέθους οπτικά συστήματα. Ένας δεύτερος σημαντικός παράγοντας είναι ότι σήμερα η CMOS τεχνολογία προσφέρει μικρά μεγέθη

(submicron sizes) και λίγες ατέλειες (low defect levels) επιτρέποντας επαρκή μεγέθη pixel (cost-effective pixel sizes) και μικρό ρεύμα διαρροής (το λεγόμενο dark current).

Τόσο τα CMOS όσο και τα CCD chips συλλαμβάνουν το φως μέσω παρόμοιων μηχανισμών εκμεταλλευόμενοι το φωτοηλεκτρικό φαινόμενο το οποίο συμβαίνει όταν τα φωτόνια αλληλεπιδρούν με το κρυσταλλοποιημένο πυρίτιο. Σημειώνουμε ότι ο όρος "CMOS" αναφέρεται στην διαδικασία με την οποία κατασκευάζεται ο αισθητήρας και όχι σε συγκεκριμένη οπτική τεχνολογία. Ηλεκτρόνια έπειτα συλλέγονται σε ένα πηγάδι (well) μέχρι η διάρκεια φωτισμού και καταγραφής να ολοκληρωθεί και έπειτα μετατρέπονται είτε σε τάση (CMOS processors) είτε μεταφέρονται σε ένα καταχωρητή μέτρησης (metering register, CCD αισθητήρας). Η μετρούμενη τάση τροφοδοτεί έπειτα ένα μετατροπέα από αναλογικό σε ψηφιακό και μετατρέπει το αναλογικό σήμα  $i_a(x,y)$  σε διακριτό  $i(n_1,n_2)$ .

$$i(n_1,n_2) = i_a(n_1T_1,n_2T_2)$$

Το κάθε ένα δείγμα  $i(n_1,n_2)$  ονομάζεται pixel και αναπαρίσταται από bits. Ένα pixel με 8-bits αναπαριστά μέχρι 256 επίπεδα φωτεινότητας. Το 0 αναπαριστά το μαύρο και το 255 το λευκό επίπεδο. Ο A/D μετατροπέας κλιμακώνει το σήμα σε επίπεδα  $i_q$ ,  $i=1,2,3\dots$  όπου  $q=2^b$ .

Η ποιότητα της εικόνας είναι συνυφασμένη με τον αριθμό των pixels που συνιστούν τον αισθητήρα, όσο περισσότερα τόσο καλύτερα. Το ποσό της λεπτομέρειας που μια κάμερα μπορεί να αιχμαλωτίσει ονομάζεται ανάλυση - ευκρίνεια (resolution) και μετράται σε pixels. Γενικά, όσο περισσότερα pixels έχει μια κάμερα τόσο περισσότερη λεπτομέρεια μπορεί να αιχμαλωτίσει. Όμως αυξάνοντας την ευκρίνεια του αισθητήρα χωρίς να αυξάνεται το μέγεθός του μειώνεται η περιοχή των μεμονωμένων pixels και συνεπώς η ευαισθησία τους. Για την αύξηση της ευαισθησίας του αισθητήρα χρησιμοποιούνται micro-lenses εστιάζοντας περισσότερο φως σε κάθε pixel. Επιπλέον, για τον υπολογισμό του aspect ratio, της αναλογίας του πλάτους και ύψους της εικόνας, διαιρούμε το μεγαλύτερο αριθμό της ανάλυσης με το μικρότερο. Για παράδειγμα αν ένας αισθητήρας έχει ανάλυση 1800x1600 τότε το aspect ratio είναι:

$$R = \frac{1800}{1600} = 1.33$$

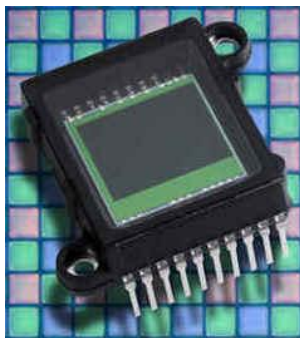
Επίσης ο ρυθμός στον οποίο τα δεδομένα διαβάζονται από έναν αισθητήρα ονομάζεται readout rate:

$$R = \frac{\text{images}}{\text{sec}} * \frac{\text{lines}}{\text{image}} * \frac{\text{pixels}}{\text{line}}$$

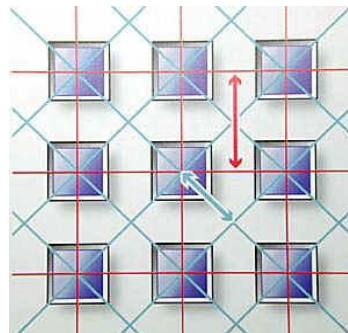
### 3.2 Διαφορές

Τόσο οι CCD όσο και οι CMOS αισθητήρες αιχμαλωτίζουν φως μέσω ενός πλέγματος μικρών εικονοστοιχείων στις επιφάνειές τους. Οι διαφορές τους όμως εντοπίζονται στον τρόπο επεξεργασίας της εικόνας και στον τρόπο κατασκευής τους. Σε κάθε περίπτωση η αρχιτεκτονική του αισθητήρα και η φυσική διάταξη (physical layout) εξαρτάται από την τεχνολογία [34], [46].

Οι CCD είναι μια συλλογή από μικροσκοπικές διόδους, ευαίσθητες στο φως, που μετατρέπουν φωτόνια (φως) σε ηλεκτρόνια.



(α): CCD Image sensor  
(Courtesy of IBM)

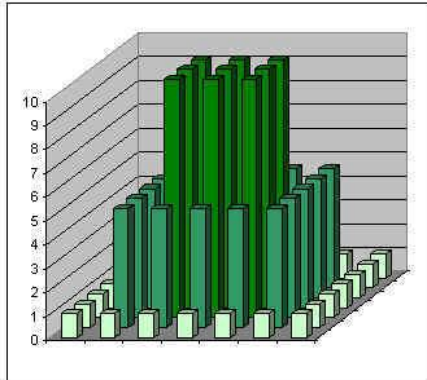
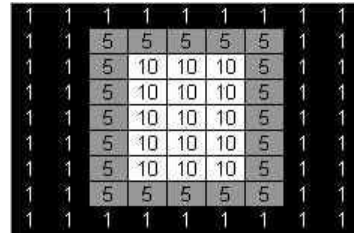


(β): Photosites τοποθετημένα  
σε γραμμές και στήλες

Σχήμα 3.2: Αισθητήρας και photosites

Αυτές οι διόδους ονομάζονται photosites. Κάθε photosite είναι ευαίσθητο στο φως και όσο πιο φωτεινό είναι το φως που χτυπά κάθε photosite, τόσο περισσότερο είναι το electrical charge που θα συγκεντρωθεί σε αυτό το σημείο. Κάθε ένα από αυτά τα photosites καταγράφει την ένταση της φωτεινότητας του φωτός που χτυπά τον αισθητήρα. Η φωτεινότητα στη συνέχεια αποθηκεύεται σαν ένα σύνολο από αριθμούς οι οποίοι μπορούν να χρησιμοποιηθούν έπειτα για να θέσουν το χρώμα και τη φωτεινότητα του κάθε pixel για την ανακατασκευή της εικόνας.

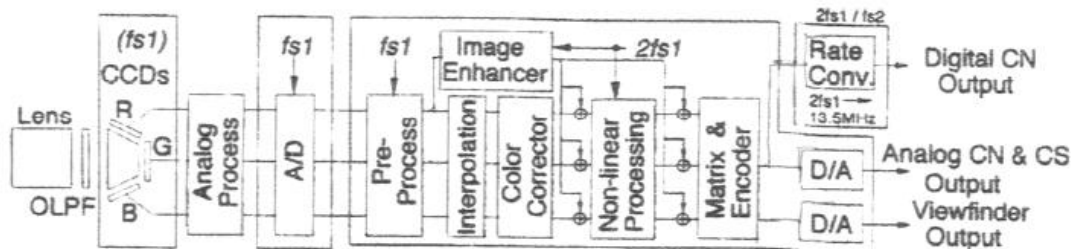
Ο αριθμός των ηλεκτρονίων που αποθηκεύεται σε κάθε photosite είναι ανάλογος της ποσότητας του φωτός που το χτυπά.



Ο αισθητήρας έχει διαφορετικό αριθμό ηλεκτρονίων στοιβαγμένα σε κάθε photosite. Τα αποθηκευμένα ηλεκτρόνια μετατρέπονται σε μια σειρά αναλογικών charges που στη συνέχεια μετατρέπονται σε ψηφιακές τιμές μέσω ενός (A/ D) μετατροπέα.

Σχήμα 3.3: Αποθήκευση ηλεκτρονίων

Στην επόμενη εικόνα απεικονίζεται το block διάγραμμα μιας ψηφιακής κάμερας που χρησιμοποιεί αισθητήρες CCDs για την σύλληψη της εικόνας.



Σχήμα 3.4: Σχηματικό Block διάγραμμα μιας έγχρωμης Video κάμερας

Ένας απλοποιημένος τρόπος για την περιγραφή ενός αισθητήρα που χρησιμοποιείται σε μια κάμερα είναι να τον σκεφθεί κανείς σαν ένα δισδιάστατο πίνακα με χίλια ή εκατομμύρια μικροσκοπικά ηλιακά κελιά, καθένα από τα οποία μετατρέπει το φως από ένα μικρό κομμάτι της εικόνας σε ηλεκτρόνια.

Το επόμενο βήμα στην διαδικασία καταγραφής της εικόνας είναι η ανάγνωση των τιμών (accumulated charge) κάθε κελιού στην εικόνα. Σε μια συσκευή CCD, η φόρτιση (charge) μεταφέρεται ουσιαστικά σε όλο το chip και διαβάζεται από την μια μεριά του πίνακα. Στη συνέχεια ένας μετατροπέας αναλογικός σε ψηφιακό μετατρέπει την τιμή

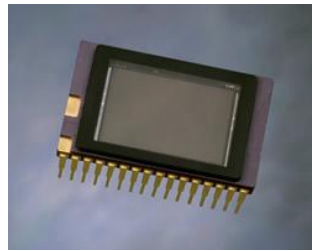
κάθε pixel στην ψηφιακή της τιμή. Στις περισσότερες CMOS συσκευές ωστόσο υπάρχουν πολλά transistors σε κάθε pixel, τα οποία ενισχύουν τη φόρτιση και τη μεταφέρουν χρησιμοποιώντας παραδοσιακά καλώδια. Η CMOS προσέγγιση επομένως είναι πιο ευέλικτη αφού κάθε pixel μπορεί να διαβάζεται αυτόνομα. Οι CCDs χρησιμοποιούν μια ειδική κατασκευαστική διαδικασία για την δημιουργία της ικανότητας να μεταφέρουν τη φόρτιση (charge) στο chip χωρίς διαστρέβλωση. Η διαδικασία αυτή οδηγεί σε υψηλής ποιότητας αισθητήρες αναφορικά με την αξιοπιστία και την ευαισθησία στο φως. Αντίθετα τα CMOS chips, χρησιμοποιούν μια κανονική διαδικασία κατασκευής για την δημιουργία του chip, η οποία χρησιμοποιείται σε όλους τους μικροεπεξεργαστές.

Εξαιτίας των κατασκευαστικών διαφορών λοιπόν υπάρχουν πολλές αξιοσημείωτες διαφορές μεταξύ των CCD και CMOS αισθητήρες. Οι CCD αισθητήρες δημιουργούν υψηλής ποιότητας και χαμηλού θορύβου εικόνες ενώ οι CMOS αισθητήρες, είναι πιο επιρρεπής στον θόρυβο. Επειδή κάθε pixel σε ένα CMOS αισθητήρα έχει πολλά transistors εντοπισμένα δίπλα σε αυτό, η ευαισθησία στο φως είναι μικρότερη αφού πολλά από τα φωτόνια που χτυπούν το chip, χτυπούν τα transistors αντί για τις φωτοδιόδους.

Οι CMOS αισθητήρες ωστόσο καταναλώνουν λιγότερη ενέργεια ενώ οι CCDs χρησιμοποιούν μια διαδικασία που καταναλώνει πολλή ενέργεια. Είναι αξιοσημείωτο ότι οι CCDs καταναλώνουν 100 φορές περισσότερη ενέργεια από έναν ισοδύναμο CMOS αισθητήρα. Τα CMOS chips μπορούν να χαλκευτούν (fabricated) σε οποιαδήποτε βασική γραμμή παραγωγής πυριτίου και έτσι τείνουν να είναι αρκετά φθηνοί συγκρινόμενοι με τους CCD αισθητήρες. Αντίθετα οι CCD αισθητήρες κατασκευαζόταν μαζικά για μεγάλο χρονικό διάστημα και έτσι είναι πιο ώριμοι και τείνουν να έχουν υψηλότερης ποιότητας pixels. Οι CCDs τείνουν να χρησιμοποιούνται σε κάμερες που εστιάζουν σε υψηλής ποιότητας εικόνες με πολλά pixels και άριστη ευαισθησία στο φως. Οι CCDs έχουν 100% fill factor, παράγοντας που δηλώνει την ευαισθησία στο φως αλλά οι CMOS έχουν πολύ λιγότερο. Οι CMOS αισθητήρες συνήθως έχουν χαμηλότερη ποιότητα, μικρότερη ανάλυση και μικρότερη ευαισθησία.

Οι κάμερες CMOS είναι πολύ φθηνότερες και έχουν μεγαλύτερη διάρκεια μπαταρίας. Με τον χρόνο οι CMOS αισθητήρες βελτιώνονται ώστε να φθάσουν να είναι σε ισότιμοι με τις CCD συσκευές στις περισσότερες εφαρμογές.

### 3.3 Οπτικοί αισθητήρες CMOS



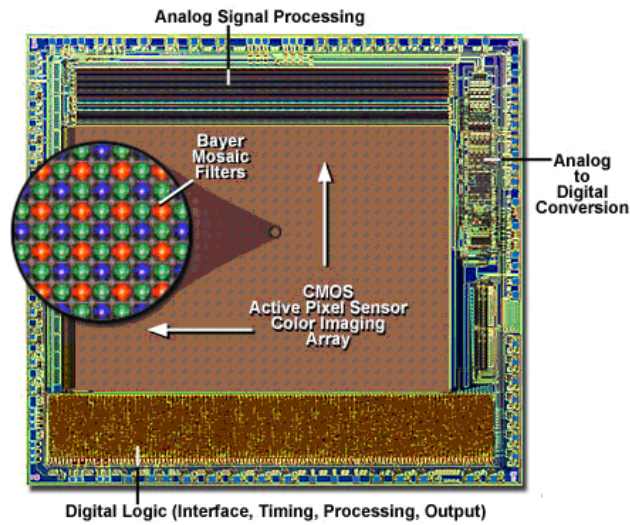
Σχήμα 3.5: CMOS αισθητήρας

Στη συνέχεια εστιάζουμε στα χαρακτηριστικά και τη λειτουργία αισθητήρων CMOS. Ένας CMOS αισθητήρας λοιπόν, χρησιμοποιεί φωτοδιόδους ή photogate transistors σαν μηχανισμό ανίχνευσης φωτός. Σε αντίθεση με τους CCD αισθητήρες, κάθε φωτο-στοιχείο διευθυνσιοδοτείται ανεξάρτητα σε ένα διάνυσμα γραμμής και στήλης, παρόμοια με μια δομή DRAM, επιτρέποντας επί μέρους ανάγνωση της εικόνας, windowing, area of interest, το οποίο είναι αρκετά χρήσιμο στην ανίχνευση κίνησης [35]. Έτσι μειώνονται οι απαιτήσεις σε μνήμη αποθήκευσης, επιτρέποντας υψηλότερα frame rates.

Έτσι κάθε pixel περιέχει, εκτός από μια φωτοδίοδο, μια τριάδα από transistors που μετατρέπουν τη συσσωρευμένη electron charge σε μια μετρήσιμη τάση, ενισχύουν την τάση για την ανάγνωση, επαναρυθμίζουν τη φωτοδίοδο, και μεταφέρουν την τάση σε μια αρτηρία.

Η αρχιτεκτονική ενός χαρακτηριστικού αισθητήρα εικόνας CMOS παρουσιάζεται στο σχήμα 6, για ένα ολοκληρωμένο κύκλωμα που περιέχει μια ενεργή περιοχή εικόνας 640 X 480 pixels. Η σειρά φωτοδίοδων, που βρίσκεται στη μεγάλη καφε-κόκκινη κεντρική περιοχή του τσιπ, καλύπτεται από ένα διαταγμένο λεπτό στρώμα κόκκινων, πράσινων, και μπλε πολυμερών φίλτρων.

## Οπτικοί Αισθητήρες



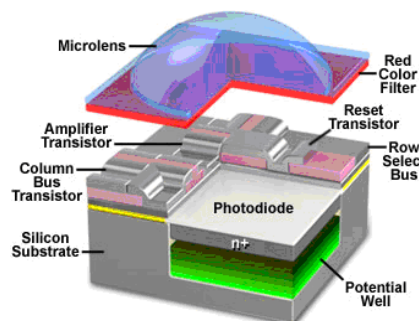
Σχήμα 3.6 Αρχιτεκτονική CMOS αισθητήρα

Στην επόμενη εικόνα απεικονίζεται η δομή ενός φίλτρου Bayer που περιγράφεται παρακάτω κάτω από το οποίο βρίσκονται οι φωτοδιόδοι που φαίνονται στην εικόνα 2γ. Μέσα στο άσπρο τετράγωνο βρίσκεται μια φωτοδιόδος και τα γράμματα T και P δηλώνουν τα transistors του pixel και τη φωτοδιόδο αντίστοιχα.



Σχήμα 3.7: (α) Φίλτρο Bayer, (β) Mosaic Array, (γ) Φωτοδιόδοι

Στο επόμενο σχήμα απεικονίζεται ένα ενεργό pixel του αισθητήρα όπου φαίνονται όσα αναφέραμε παραπάνω.



Σχήμα 3.8: Ενεργό pixel του αισθητήρα

Όσον αφορά την ποιότητα της εικόνας των σύγχρονων CMOS αισθητήρων, σε σχέση με αυτή των συσκευών CCD, παρουσιάζει μεγαλύτερα επίπεδα θορύβου, οφειλόμενα κυρίως στην ενσωμάτωση αναλογικών και ψηφιακών κυκλωμάτων στο ίδιο chip. Οι CMOS ωστόσο δεν υποφέρουν από blooming και smearing, επιπτώσεις που προκαλούνται από διαρροή ρεύματος στις συσκευές CCD.

Οι CMOS οπτικοί αισθητήρες αποτελούν την καλύτερη επιλογή για εφαρμογές που απαιτούν χαμηλή κατανάλωση ενέργειας, μικρό μέγεθος και μέση ανάλυση εικόνας. Τέτοιες εφαρμογές είναι τα βιομετρικά συστήματα (π.χ. η αναγνώριση προσώπου) και οι automotive εφαρμογές (συστήματα οδηγικής βοήθειας). Ενσωματώνουν τα αναλογικά και ψηφιακά μέρη σε ένα chip, παρέχοντας άμεση διεπαφή στο data bus του επεξεργαστή και έτσι μειώνουν σημαντικά τον αριθμό των συστατικών μερών, την κατανάλωση ενέργειας και το μέγεθος του συνολικού board συγκρινόμενο με παραδοσιακές επιλογές.

Ένα παράδειγμα χρήσης οπτικών αισθητήρων είναι στα κινητά τηλέφωνα και στα λεγόμενα Camera Phones, όπου η κατανάλωση ισχύος αποτελεί σημαντική ανησυχία, αφού οι καταναλωτές επιθυμούν τα τηλέφωνα τους να λειτουργούν για μεγάλο διάστημα μεταξύ των φορτίσεων, καθώς και να έχουν μικρό βάρος το οποίο σημαίνει μικρές μπαταρίες. Σαν αποτέλεσμα οι περισσότεροι σχεδιαστές Camera Phone επιλέγουν τη χρήση CMOS οπτικοί αισθητήρες αντί για τη CCD τεχνολογία, επειδή η τεχνολογία των CMOS αισθητήρων καταναλώνει λιγότερη ισχύ.

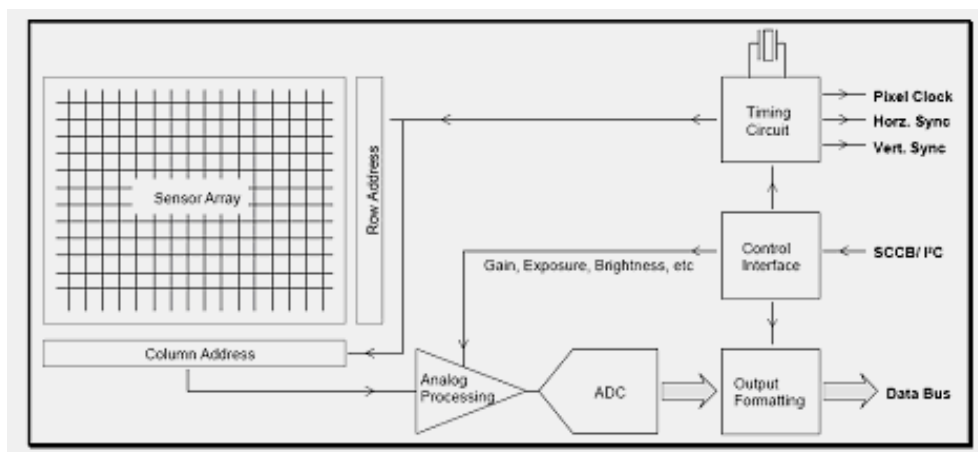
Η τεχνολογία CCD απαιτεί πολλαπλά chips και πολλαπλές τάσεις τροφοδοσίας για να υλοποιηθεί. Αντίθετα οι CMOS αισθητήρες μπορούν να παρέχουν επιπρόσθετη εξοικονόμηση ενέργειας επιτελώντας επιπρόσθετη επεξεργασία, όπως pixel interpolation και color space conversion, πάνω στο chip μειώνοντας έτσι την κατανάλωση. Επιπλέον οι CMOS αισθητήρες χρειάζονται χαμηλό ρεύμα για τη λειτουργία της αναμονής μειώνοντας την ανάγκη για περικοπή ισχύος στα κυκλώματα της κάμερας

Οι αισθητήρες CMOS μπορούν να μειώσουν το κόστος του συστήματος, εξαλείφοντας συστατικά μέρη για τη μετατροπή από αναλογικό σε ψηφιακό, παραγωγή του ρολογιού, pixel interpolation και μετατροπή color space conversion, απευθείας LCD απεικόνιση, επεξεργασία εικόνας με χρήση αλγορίθμων (AWB, AGC, AEC, etc.), και power conditioning. Ο αισθητήρας μπορεί να παρέχει έξοδο 8 ή 10 bit raw data και να παρέχει χαρακτηριστικά για τον έλεγχο της ποιότητας της εικόνας, όπως: automatic



white balance (AWB) control, automatic exposure control (AEC), automatic gain control (AGC), formatted YUV data output, κ.τ.λ..

Ο αισθητήρας πρέπει να επιτρέπει την εξωτερική επεξεργασία των εικόνων ή έλεγχο αυτών των λειτουργιών ενώ παράγει την έξοδο. Ο επεξεργαστής εικόνας και οι αλγόριθμοι που υποστηρίζονται έχουν σημαντική επίπτωση στην ποιότητα της εικόνας. Πολλοί αλγόριθμοι υπάρχουν για να επιτελούν exposure control, white balance, pixel interpolation και όσο πιο δυναμικοί και επαρκείς είναι οι αλγόριθμοι αυτοί και το υλικό DSP, τόσο καλύτερη είναι η απόδοση και η ποιότητα της εικόνας. Οι CMOS αισθητήρες έχουν το πλεονέκτημα ότι μπορούν εύκολα να σχεδιαστούν αφού ενσωματώνουν τον A/D μετατροπέα, τη διαχείριση ενέργειας και πρόσθετα υποστηρικτικά κυκλώματα. Παρακάτω φαίνεται ένα κύκλωμα αισθητήρα CMOS.



**Σχήμα 3.9:** Σύστημα οπτικού αισθητήρα CMOS

Οι έγχρωμοι αισθητήρες περιλαμβάνουν ένα χρωματικό φίλτρο Bayer και ο αναλογικός προ-επεξεργαστής συμπληρώνεται με κυκλώματα gamma correction και κυκλώματα μετατροπής RGB σε YCrCb. Τα δεδομένα της εικόνας συνήθως μεταδίδονται με μια παράλληλη διεπαφή bus ενώ ο CMOS αισθητήρας τυπικά ελέγχεται από το I<sup>2</sup>C πρωτόκολλο.

Ανάλογα με τον αισθητήρα ο host επεξεργαστής μπορεί να ελέγχει το exposure, το κέρδος (gain), τη φωτεινότητα (brightness), την αντίθεση (contrast), το gamma correction, την παραθύρωση (windowing), το frame rate και τον τύπο της εξόδου. Εκτός από την αρχιτεκτονική progressive-scan, η οποία παράγει τα δεδομένα σειριακά γραμμής-

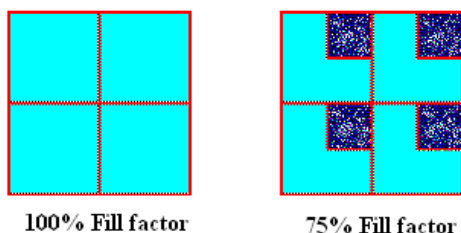
γραμμή, υπάρχουν οι random-access αισθητήρες οι οποίοι συνδέονται στον host μέσω διευθύνσεων γραμμών και στηλών παρόμοια με μια DRAM συσκευή. Οι αισθητήρες αυτοί παρέχουν πλήρη πρόσβαση σε οποιοδήποτε pixel μέσα στο πίνακα και είναι εξαιρετικά χρήσιμοι για συστήματα ανίχνευσης κίνησης.

Οι CMOS αισθητήρες υπάρχουν σε διάφορες αναλύσεις. Τυπικές είναι οι CIF (352x288 pixels), VGA (640x480), SVGA (800x600) και XGA (1024x768) συσκευές. Οι χαμηλής ανάλυσης αισθητήρες (128x128) επιτρέπουν εκατοντάδες frames per second και χρησιμοποιούνται για γρήγορη ανίχνευση κίνησης, ενώ άλλα συστήματα απαιτούν μεγαλύτερες αναλύσεις μέχρι κάποια Megapixels.

Ένα από τα κύρια μειονεκτήματα είναι ο θόρυβος, γνωστός σαν fixed pattern noise (FPN). Διακυμάνσεις στο κέρδος και στο offset των transistor, που αποτελεί σημαντικό πρόβλημα στη διαδικασία κατασκευής της τεχνολογίας CMOS, δημιουργούν προβλήματα στην απόδοση των transistor. Το αποτέλεσμα είναι θόρυβος εμφανής στις συλληφθείσες εικόνες που είναι σταθερός και αναπαραγωγίμος από τη μια εικόνα στην άλλη. Στις περισσότερες περιπτώσεις, ο θόρυβος αυτός μπορεί να μειωθεί σημαντικά συντονίζοντας τα αναλογικά κυκλώματα που βρίσκονται στην περιφέρεια του πίνακα.

Εξαιτίας του μεγαλύτερου επίπεδου θορύβου των CMOS αισθητήρων, ο χρόνος επεξεργασίας μεταξύ των εικόνων είναι μεγαλύτερος, επειδή οι αισθητήρες αυτοί χρησιμοποιούν ψηφιακή επεξεργασία (DSP) για την μείωση και απαλοιφή του θορύβου.

Επιπλέον, ενώ οι CMOS αισθητήρες καταγράφουν εξαιρετικές εικόνες σε ηλιόλουστες ημέρες ωστόσο αντιμετωπίζουν πρόβλημα σε καταστάσεις χαμηλού φωτισμού. Η ευαισθησία τους μειώνεται γιατί τμήμα του κάθε photosite καταλαμβάνεται από κυκλώματα που φιλτράρουν τον θόρυβο και επιτελούν άλλες λειτουργίες. Το ποσοστό των pixel που είναι αφιερωμένα στην συλλογή φωτός ονομάζεται fill factor [46].



Σχήμα 3.10: Fill factor

Όσο μικρότερος ο παράγοντας fill factor, τόσο λιγότερο ευαίσθητος είναι ο αισθητήρας και τόσο μεγαλύτερος χρόνος έκθεσης απαιτείται. Για την αντιμετώπιση του προβλήματος προστίθενται φακοί micro-lenses σε κάθε pixel για τη συγκέντρωση φωτός και την εστίαση σε κάθε photosite.

### 3.4 Ο αισθητήρας επιλογής μας

#### 3.4.1 Γενικά Χαρακτηριστικά

Στη συνέχεια περιγράφουμε ένα σύγχρονο αισθητήρα, τον οποίο επιλέξαμε να χρησιμοποιήσουμε στο σύστημά μας λαμβάνοντας υπόψη και τα παραπάνω, της εταιρείας Micron, μιας από τις μεγαλύτερες εταιρείες κατασκευής τέτοιων συστημάτων. Ο αισθητήρας MT9V112 είναι ένας VGA CMOS αισθητήρας τελευταίας τεχνολογίας [36]. Η συσκευή αυτή περιέχει, εκτός των άλλων έναν τέταρτης γενιάς επεξεργαστή εικόνας (digital image flow processor) και συλλαμβάνει υψηλής ποιότητας εικόνες VGA ανάλυσης. Έχει το πλεονέκτημα χρήσης ειδικής τεχνολογίας της Micron, η οποία επιτυγχάνει να συλλαμβάνει εικόνες χαμηλού θορύβου διατηρώντας τα πλεονεκτήματα της CMOS τεχνολογίας.

Ο αισθητήρας αποτελεί μια ολοκληρωμένη camera-on-a-chip λύση που συνδυάζει χαμηλή κατανάλωση και χαμηλό κόστος, ικανοποιώντας τις απαιτήσεις των φορητών συσκευών όπως κινητά τηλέφωνα, PDAs, και παιχνίδια. Ενσωματώνει εξελιγμένες λειτουργίες on-chip και προγραμματίζεται μέσω ενός απλού σειριακού πρωτοκόλλου. Ανάμεσα στις λειτουργίες που προσφέρει είναι το color recovery, color correction, sharpening, programmable gamma correction, auto black reference clamping, auto exposure, automatic 50Hz/60Hz flicker avoidance, lens shading correction, auto white balance (AWB) και on-the-fly defect identification and correction.

Ο MT9V112 μπορεί να προγραμματιστεί για να παράγει progressive-scan εικόνες επιτυγχάνοντας ρυθμούς που φθάνουν τα 30 frames per second (fps). Τα δεδομένα της εικόνας μπορούν να παραχθούν σε μια από τις 8-bit μορφές:

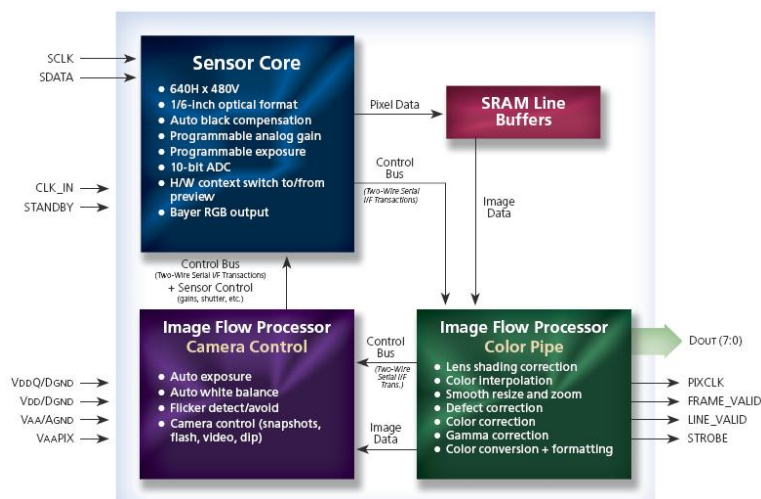
- ITU-R BT.656 (CCIR656, progressive scan only) YCbCr
- 565RGB
- 555RGB

- 444RGB
- Raw Bayer
- Processed Bayer

Οι εικόνες βγαίνουν στην έξοδο από την αρτηρία dout μεγέθους 8-bit. Χρησιμοποιούνται δύο ρολόγια. Το pixclk χρησιμοποιείται για να συγχρονίζει την έξοδο, μαζί με τα σήματα line\_valid και frame\_valid που δηλώνουν έγκυρα δεδομένα και παράγονται από τον ίδιο τον αισθητήρα. Το pixclk παράγει ρολόι με τη μισή συχνότητα του ρολογιού εισόδου. Ο MT9V112 δέχεται ρολόι εισόδου με συχνότητες μέχρι τα 27MHz, επιτυγχάνοντας ρυθμούς 30 fps για εικόνες ανάλυσης VGA. Για να λειτουργήσει απαιτεί μια παροχή παροχής ηλεκτρικού ρεύματος, ενός φακού (lens), και ενός ρολογιού. Επίσης έχει τη δυνατότητα να τίθεται σε κατάσταση αναμονής.

### 3.4.2 Αρχιτεκτονική

Εσωτερικά ο MT9V112 αποτελείται από ένα πυρήνα ( sensor core) και έναν επεξεργαστή εικόνας (image flow processor - IFP). Ο IFP διακρίνεται σε δυο τμήματα: το colorpipe (CP), και τον ελεγκτή κάμερας (camera controller - CC).



Σχήμα 3.11: Λειτουργικό Block Διάγραμμα

Ο πυρήνας του αισθητήρα συλλαμβάνει εικόνες σε μορφή raw Bayer οι οποίες στην συνέχεια αποτελούν είσοδο στο IFP. Το τμήμα CP του IFP επεξεργάζεται το εισερχόμενο stream για να δημιουργήσει interpolated, color-corrected έξοδο ενώ το τμήμα CC ελέγχει τον sensor core για να διατηρήσει το επιθυμητό exposure και color

balance και να υποστηρίξει snapshot modes. Οι καταχωρητές προγραμματίζονται μέσω ενός σειριακού πρωτοκόλλου επικοινωνίας.

Όσον αφορά τη slave διεύθυνση του αισθητήρα που χρησιμοποιείται από το πρωτόκολλο επικοινωνίας, η διεύθυνση εγγραφής είναι 0XBA ενώ η διεύθυνση ανάγνωσης είναι 0XBB. Ο MT9V112 υποστηρίζει ένα εύρος χρωματικών τύπων προερχόμενοι από τις 4 βασικές αναπαραστάσεις χρώματος: YCbCr, RGB, raw Bayer (μη επεξεργασμένο, απευθείας από τον αισθητήρα), και επεξεργασμένο Bayer (δεδομένα τύπου Bayer προερχόμενα από επεξεργασμένο RGB).

### 3.4.3 Έλεγχος κάμερας

Ο ελεγκτής της κάμερας συνεχώς συγκεντρώνει την φωτεινότητα της εικόνας και τα στατιστικά των χρωμάτων. Δυο τμήματα χρησιμοποιούν αυτές τις μετρήσεις για να ρυθμίσουν τον αισθητήρα. Η μονάδα auto exposure ρυθμίζει το gain και το shutter-width. Ο χρήστης μπορεί να ελέγξει την ταχύτητα και την ευαισθησία του αλγορίθμου. Η μονάδα AWB ρυθμίζει τα gains και το CCM και ο χρήστης μπορεί να ελέγξει την περιοχή της σκηνής που πρόκειται να αναλυθεί και την ανταπόκριση του αλγορίθμου στις αλλαγές φωτεινότητας.

Ο MT9V112 παράγει δεδομένα σε μορφή ITU-R BT.656 η οποία ουσιαστικά περιέχει YCbCr 4:2:2 δεδομένα και είναι κατάλληλη για JPEG/MPEG συμπίεση και για αναπαράσταση από συσκευές video (progressive scan).

MODE	BYTE			
Default	Cbi	Yi	Cri	Yi+1
Swap CrCb	Cri	Yi	Cbi	Yi+1
SwapYC	Yi	Cbi	Yi+1	Cri
Swap CrCb, SwapYC	Yi	Cri	Yi+1	Cbi

**Πίνακας 3.1:** Έξοδος σε YCbCr μορφή

Επίσης έχει τη δυνατότητα για παραγωγή της εξόδου σε μια από τις μορφές RGB η οποία είναι κατάλληλη για LCD συσκευές. Ο αισθητήρας λοιπόν μπορεί να

## Οπτικοί Αισθητήρες

προγραμματίζεται να παράγει 16-bit RGB (RGB565), 15-bit RGB (RGB555) και 2 τύπους 12-bit RGB (RGB444). Στους επόμενους πίνακες φαίνονται οι μορφές εξόδου RGB.

MODE	LINE	BYTE			
Default	First	Gi	Ri+1	Gi+2	Ri+3
	Second	Bi	Gi+1	Bi+2	Gi+3
Flip Bayer Col	First	Ri	Gi+1	Ri+2	Gi+3
	Second	Gi	Bi+1	Gi+2	Bi+3
Flip Bayer Row	First	Bi	Gi+1	Bi+2	Gi+3
	Second	Gi	Ri+1	Gi+2	Ri+3
Flip Bayer Col, Flip Bayer Row	First	Gi	Bi+1	Gi+2	Bi+3
	Second	Ri	Gi+1	Ri+2	Gi+3

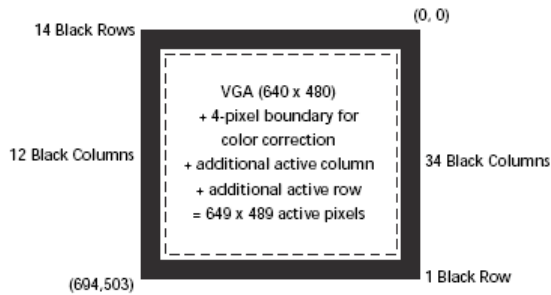
**Πίνακας 3.2:** Έξοδος σε επεξεργασμένη Bayer Μορφή

MODE (SWAP DISABLED)	BYTE	D7	D6	D5	D4	D3	D2	D1	D0
RGB565	First	R7	R6	R5	R4	R3	G7	G6	G5
	Second	G4	G3	G2	B7	B6	B5	B4	B3
RGB555	First	0	R7	R6	R5	R4	R3	G7	G6
	Second	G5	G4	G3	B7	B6	B5	B4	B3
RGB444x	First	R7	R6	R5	R4	G7	G6	G5	G4
	Second	B7	B6	B5	B4	0	0	0	0
RGBx444	First	0	0	0	0	R7	R6	R5	R4
	Second	G7	G6	G5	G4	B7	B6	B5	B4

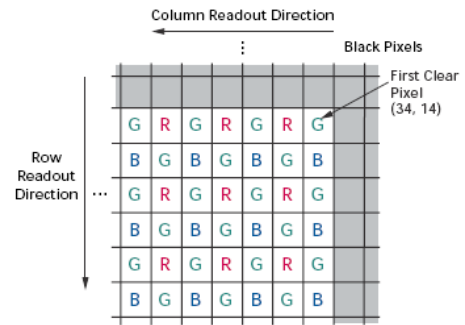
**Πίνακας 3.3:** Έξοδος σε RGB μορφή

Ο πυρήνας του αισθητήρα αποτελείται από ένα πίνακα pixel μεγέθους 695 x 504 , μια αλυσίδα αναλογικής ανάγνωσης, 10-bit ADC με προγραμματιζόμενο κέρδος, ρολόι και έλεγχο. Στην επόμενη εικόνα απεικονίζεται το μέγεθος του πίνακα των pixel ο οποίος περιέχει 649 στήλες και 489 γραμμές ενεργών pixel και το σχέδιο των χρωμάτων που ουσιαστικά είναι στη μορφή Bayer.

## Οπτικοί Αισθητήρες

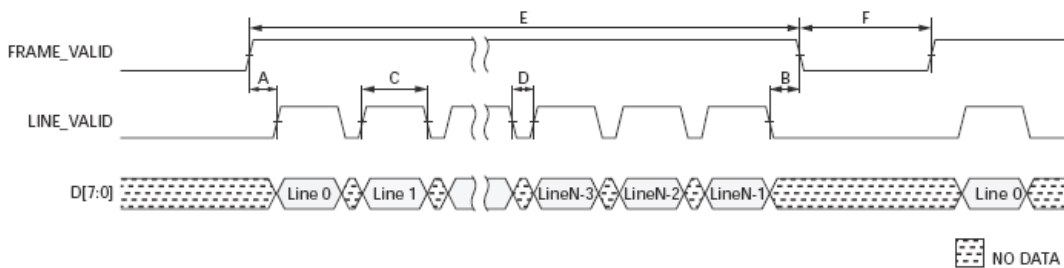


**Σχήμα 3.12:** Πίνακας pixel

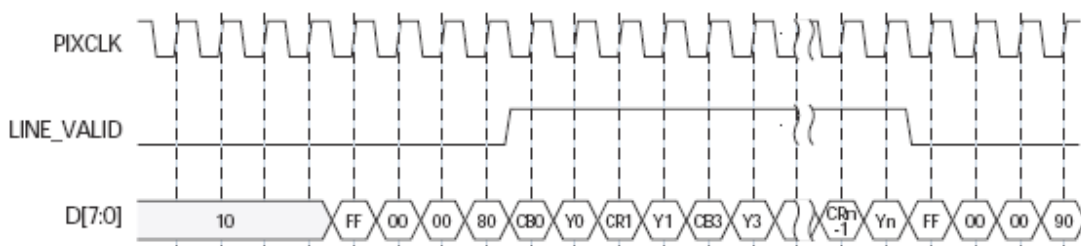


**Σχήμα 3.13:** Σχέδιο χρωμάτων pixel

Παρακάτω απεικονίζεται ο χρονισμός για το vertical, το frame\_valid γίνεται 0 στο τέλος του πλαισίου και horizontal timing όπου το line\_valid γίνεται 0 στο τέλος κάθε γραμμής..



**Σχήμα 3.14:** Vertical timing



**Σχήμα 3.15:** Horizontal timing

Ο αισθητήρας τέλος δίνει τη δυνατότητα για κατάσταση αναμονής όπου η κατανάλωση ενέργειας είναι ελάχιστη ενώ όλα τα εσωτερικά ρολόγια σταματούν. Επιπλέον ο αισθητήρας χρησιμοποιεί 2 κύρια ρολόγια, ένα ρολόι (CLKIN σήμα ) που αποτελεί είσοδο από το master σύστημα που ελέγχει τον αισθητήρα (διεπαφή I<sub>2</sub>C) και στην προκειμένη περίπτωση είναι το σύστημα που υλοποιήσαμε και ένα pixel ρολόι,

pixel clock σήμα, το οποίο συγχρονίζει την έξοδο της εικόνας και τρέχει στη μισή συχνότητα του ρολογιού του master.

### **3.5 Συμπεράσματα**

Οι αισθητήρες CMOS χρησιμοποιούνται ολοένα και περισσότερο σε περισσότερες εφαρμογές. Μακροχρόνιες έρευνες κατόρθωσαν να βελτιώσουν την ποιότητα των εικόνων που καταγράφονται από αισθητήρες CMOS και να είναι πλέον συγκρίσιμη με αυτή των CCD. Οι οπτικοί αισθητήρες CMOS έχουν το πλεονέκτημα της ενσωμάτωσης πολλών κυκλωμάτων στο ίδιο chip μειώνοντας τον αριθμό των chip που χρειάζονται για ένα CCD αισθητήρα και επιτρέποντας την ενσωμάτωση πολλών νέων λειτουργιών με μειωμένο κόστος. Έχουν ήδη χρησιμοποιηθεί σε μηχανήματα fax, σαρωτές, κάμερες ασφαλείας, παιχνίδια με μεγάλη επιτυχία και συνεχίζουν να ενσωματώνονται σε ποικίλες εφαρμογές. Εξαιτίας της ικανότητάς τους να καταγράφουν εικόνες σε υψηλούς ρυθμούς υπάρχει μεγάλο ενδιαφέρον για τη χρήση τους σε οπτικά συστήματα και διαγνωστικές συσκευές. Ωστόσο δεν έχουν αντικαταστήσει ακόμα τους CCD αισθητήρες σε εφαρμογές που απαιτούν υψηλή ποιότητα εικόνας. Γίνονται προσπάθειες όμως αντιμετώπισης των προβλημάτων μέσω της αύξησης της ποιότητας και της μείωσης του θορύβου, με αποτέλεσμα την ολοένα και συχνότερη ενσωμάτωση αυτών των αισθητήρων σε νέες συσκευές.





# 4

## Πρωτόκολλο Επικοινωνίας

### I<sup>2</sup>C (Inter-Integrated Circuit)

Στο παρόν κεφάλαιο περιγράφεται το πρωτόκολλο επικοινωνίας I<sup>2</sup>C το οποίο χρησιμοποιείται για τον προγραμματισμό των καταχωρητών του αισθητήρα. Αναφέρονται τα χαρακτηριστικά και ο τρόπος λειτουργίας και προσδιορίζονται τα πλεονεκτήματά του. Η μεταφορά δεδομένων περιγράφεται με σχηματικά παραδείγματα για την καλύτερη κατανόηση.

## 4.1 Εισαγωγή

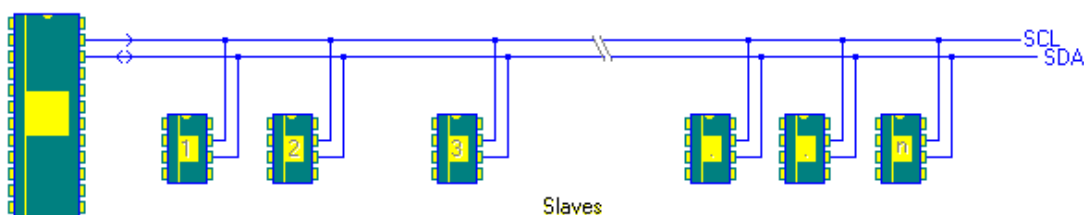
Το I<sup>2</sup>C είναι ένα πρωτόκολλο επικοινωνίας το οποίο χρησιμοποιείται για την ανταλλαγή δεδομένων μεταξύ συσκευών [38], [48]. Θεμελιώθηκε από τη Philips και σήμερα αποτελεί το παγκόσμιο de-facto standard το οποίο έχει υλοποιηθεί σε περισσότερα από 1000 ολοκληρωμένα κυκλώματα ενώ έχει δοθεί άδεια χρήσης σε περισσότερες από 50 εταιρείες ανά τον κόσμο. Αρχικός σκοπός της ανάπτυξής του ήταν ο εσωτερικός έλεγχος των ολοκληρωμένων κυκλωμάτων, από όπου προκύπτει και το όνομά του, Inter IC ή I<sup>2</sup>C.

Το I<sup>2</sup>C προσφέρει πολλά πλεονεκτήματα, τόσο από την άποψη του σχεδιαστή, όσο και από την άποψη του κατασκευαστή. Το απλό σειριακό I<sup>2</sup>C bus με τις δύο γραμμές ελαχιστοποιεί τις διασυνδέσεις και έτσι τα ολοκληρωμένα κυκλώματα έχουν λιγότερα pins, έχοντας σαν αποτέλεσμα μικρότερες και λιγότερο ακριβές PCBs. Επιπλέον δεν χρειάζεται η υλοποίηση του πρωτοκόλλου και στις δύο συσκευές που επιθυμούν να επικοινωνήσουν αφού η μια από αυτές θα έχει ήδη υλοποιημένο το πρωτόκολλο I<sup>2</sup>C.

Σε ένα σύστημα απαιτείται η μείωση του κόστους σύνδεσης των συσκευών ενώ τα κυκλώματα που εκτελούν λειτουργίες έλεγχου του συστήματος δεν χρειάζεται να έχουν μεγάλες ταχύτητες μεταφοράς δεδομένων. Έτσι δημιουργήθηκε το σειριακό I<sup>2</sup>C bus για να ικανοποιήσει αυτές τις ανάγκες. Οι κύριες εφαρμογές του πρωτοκόλλου αυτού είναι στα τηλεπικοινωνιακά συστήματα και την ψηφιακή τηλεόραση.

## 4.2 Χαρακτηριστικά

Κάθε συσκευή που συνδέεται στο I<sup>2</sup>C bus έχει μια μοναδική διεύθυνση με την οποία αναγνωρίζεται από το σύστημα. Επίσης υπάρχουν απλές master/slave σχέσεις μεταξύ των συσκευών. Κάθε συσκευή μπορεί να λειτουργεί σαν receiver ή transmitter. Ο master είναι αυτός που αρχικοποιεί μια επικοινωνία και ουσιαστικά την ελέγχει παράγοντας τους παλμούς του ρολογιού. Το I<sup>2</sup>C είναι ένα multi-master bus, που σημαίνει ότι είναι δυνατόν να ελέγχουν το bus περισσότερες από μια συσκευές και επομένως να συνδέονται σε αυτό ταυτόχρονα.



Σχήμα 4.1: Απλή υλοποίηση I<sup>2</sup>C

Αποτελείται από δύο γραμμές, την SDA η οποία είναι υπεύθυνη για την μεταφορά των δεδομένων και την SCL η οποία είναι ένα σειριακό ρολόι. Και οι δύο γραμμές είναι αμφίδρομες. Τα δεδομένα είναι έγκυρα όταν το ρολόι είναι High ενώ κάθε συσκευή αναγνωρίζεται από μια μοναδική διεύθυνση. Όταν και οι δύο γραμμές είναι High τότε το bus είναι ελεύθερο. Τα δεδομένα μεταδίδονται με ταχύτητες που φθάνουν τα 100 kbits/sec στο Standard mode, τα 400 kbits/sec στο Fast mode και τα 3,4 Mbits/sec στο High speed mode.

Δυο σημαντικές συνθήκες στην διαδικασία επικοινωνίας είναι οι Start και Stop συνθήκες οι οποίες δημιουργούνται πάντα από τον master. Η κατάσταση start δηλώνεται από μια High to Low μετάβαση ενώ η κατάσταση stop από μια Low to High μετάβαση στην SDA γραμμή. Μετά από ένα start το bus θεωρείται ότι είναι κατειλημμένο ενώ απελευθερώνεται μετά από ένα stop. Το bus συνεχίζει να είναι κατειλημμένο εάν ο master προκαλέσει κι άλλο start το λεγόμενο repeated start αντί για stop.

Start		Ο Master για την παραγωγή Start πρώτα θέτει την γραμμή SDA low και μετά θέτει την γραμμή SCL low. Μετάβαση High to Low
Stop		Ο Master πρώτα θέτει το SCL High και μετά θέτει το SDA High. Μετάβαση Low to High

Σχήμα 4.2: Start - Stop σήμα

### 4.3 Μεταφορά δεδομένων

Κάθε byte που στέλνεται από τη γραμμή SDA πρέπει να είναι 8 bit. Ο αριθμός των byte που μπορεί να μεταδοθεί σε κάθε μεταφορά είναι απεριόριστος ενώ κάθε byte ακολουθείται από ένα σήμα επιβεβαίωσης (Acknowledge bit). Επιπλέον τα δεδομένα μεταδίδονται με το πιο σημαντικό ψηφίο πρώτα (MSB). Στην περίπτωση που ύστερα από την μετάδοση ενός byte, ο slave δεν μπορεί να παραλάβει ή να αποστείλει άλλα δεδομένα πριν τελειώσει μια εσωτερική του λειτουργία, κάνει το SCL LOW ώστε ο master να μπει σε μια κατάσταση αναμονής. Η μεταφορά συνεχίζεται όταν ο slave είναι έτοιμος για άλλο byte και απελευθερώνει την γραμμή του ρολογιού (SCL) [48].

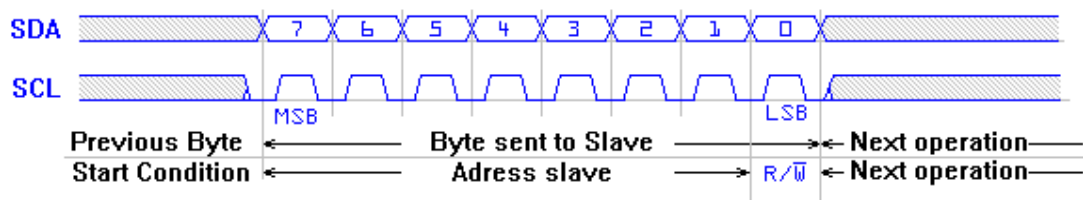
Ο παλμός του ρολογιού που σχετίζεται με το σήμα επιβεβαίωσης δημιουργείται από τον master. Ο transmitter απελευθερώνει την γραμμή SDA (High) κατά την διάρκεια του παλμού επιβεβαίωσης ενώ ο receiver πρέπει να κατεβάσει την SDA ώστε να παραμείνει σταθερά Low κατά την διάρκεια της High περιόδου του παλμού επιβεβαίωσης. Συνήθως ο receiver είναι υποχρεωμένος να δημιουργεί επιβεβαίωση μετά από τη λήψη κάποιου byte. Επίσης ο slave πρέπει να επιβεβαιώνει την διεύθυνση του όταν την στείλει ο master, ενώ εάν δεν μπορεί να δεχτεί ή να στείλει δεδομένα εξαιτίας της επεξεργασίας κάποιας εσωτερικής λειτουργίας και δεν επιβεβαιώσει την διεύθυνση τότε η γραμμή δεδομένων πρέπει να διατηρηθεί High από τον slave. Ο master μπορεί να δημιουργήσει stop για να σταματήσει την μεταφορά ή repeated start για να αρχίσει νέα μεταφορά.

Στην περίπτωση που ο slave receiver επιβεβαιώσει την διεύθυνση αλλά κάποια στιγμή κατά την διάρκεια της μεταφοράς δεν μπορεί να λάβει άλλα δεδομένα ο master πρέπει και πάλι να σταματήσει την μεταφορά. Αυτό γίνεται με τον slave να δημιουργεί not-acknowledge στο πρώτο byte που έρχεται. Ο slave διατηρεί την data line High και ο master δημιουργεί Stop ή repeated Start.

Στην περίπτωση που έχουμε master receiver τότε αυτός θα πρέπει να σηματοδοτήσει το τέλος των δεδομένων στον slave-transmitter δημιουργώντας σήμα μη επιβεβαίωσης στο τελευταίο μεταδιδόμενο byte. Ο slave transmitter πρέπει να απελευθερώσει την γραμμή δεδομένων, για να επιτρέψει στον master να δημιουργήσει σήμα τερματισμού, Stop, ή επαναλαμβανόμενης έναρξης, repeated Start.

Για την αποφυγή συγκρούσεων το πρωτόκολλο παρέχει τη λειτουργία ‘arbitration’, επιτρέποντας σε πολλούς master να επικοινωνήσουν ταυτόχρονα χωρίς προβλήματα.

Μια τυπική μεταφορά αρχίζει με τον master να στέλνει ένα start bit (μετάβαση HIGH σε LOW του sda). Μετά το start bit ο master στέλνει την διεύθυνση της slave συσκευής μεγέθους 8-bit. Το τελευταίο bit προσδιορίζει αν η αίτηση ήταν για ανάγνωση, οπότε είναι ίσο με το μηδέν, ή για εγγραφή της συσκευής, οπότε είναι ίσο με τη μονάδα. Η slave συσκευή επιβεβαιώνει την διεύθυνσή της στέλνοντας ένα bit επιβεβαίωσης στον master.

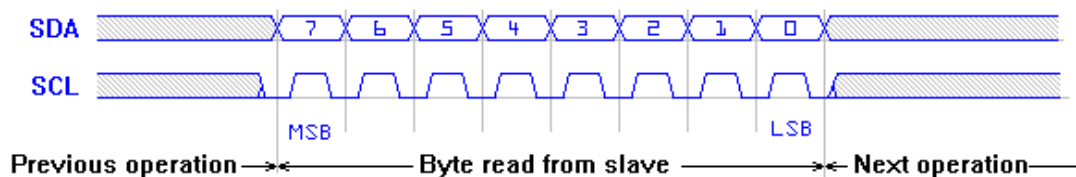


Σχήμα 4.3: Μεταφορά της διεύθυνσης της slave συσκευής [48]

Εάν η αίτηση ήταν εγγραφή, ο master μεταφέρει την διεύθυνση του καταχωρητή στον οποίο επιθυμεί να γράψει. Η slave συσκευή επιβεβαιώνει την διεύθυνσή του καταχωρητή, στέλνοντας ένα bit επιβεβαίωσης στον master. Ο αισθητήρας MT9V112 που επιλέξαμε και παρουσιάσαμε χρησιμοποιεί 16-bit για να γράψει έναν εσωτερικό καταχωρητή, οπότε απαιτούνται να γίνουν δύο μεταφορές για να ολοκληρωθεί η εγγραφή. Με την ολοκλήρωση της εγγραφής η διεύθυνση του καταχωρητή στον αισθητήρα αυξάνεται αυτόματα και έτσι τα επόμενα 16-bits γράφονται στον επόμενο καταχωρητή. Η μεταφορά σταματά όταν ο master στείλει start ή stop σήμα.

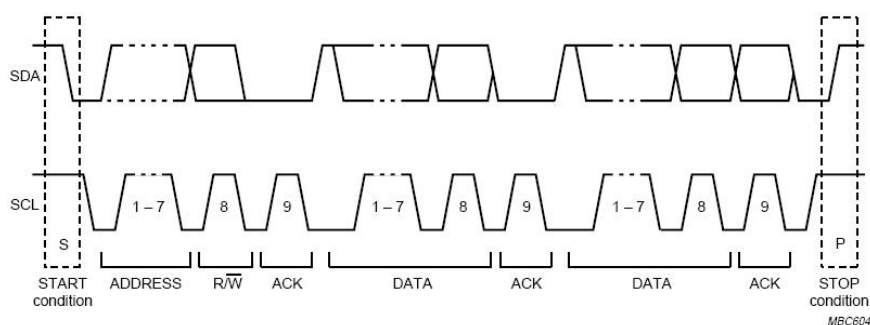
Μια τυπική ανάγνωση εκτελείται ως εξής. Ο master στέλνει την διεύθυνση εγγραφής της συσκευής και την διεύθυνση του καταχωρητή που επιθυμεί να αναγνώσει και στη συνέχεια στέλνει ένα start bit και τη διεύθυνση ανάγνωσης της συσκευής. Ο master μετά τη λήψη 8-bit στέλνει σήμα επιβεβαίωσης. Η διεύθυνση του καταχωρητή αυξάνεται αυτόματα μετά την ολοκλήρωση μεταφοράς των 16-bit. Η μεταφορά σταματά όταν ο master στείλει ένα σήμα μη επιβεβαίωσης.

## Πρωτόκολλο επικοινωνίας I<sup>2</sup>C



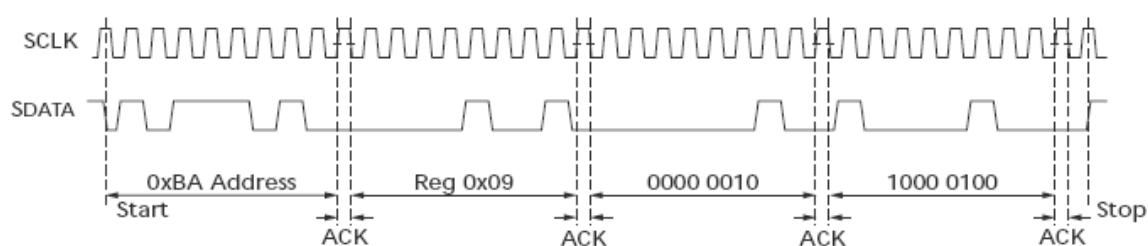
Σχήμα 4.4: Λήψη δεδομένων απο τη slave συσκευή

Μια πλήρη μεταφορά σύμφωνα με το I<sup>2</sup>C πρωτόκολλο απεικονίζεται στο επόμενο σχήμα.



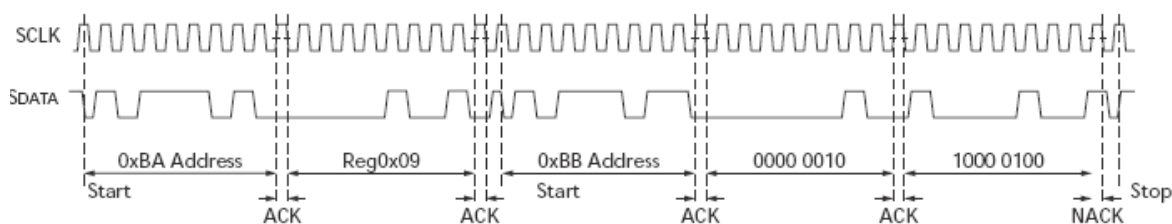
Σχήμα 4.5: Μεταφορά δεδομένων

Στα επόμενα σχήματα απεικονίζονται οι διαδικασίες εγγραφής και ανάγνωσης κάποιου καταχωρητή του αισθητήρα. Παρατηρούμε ότι σε αντίθεση με το I<sup>2</sup>C πρωτόκολλο υπάρχει μια ενδιάμεση κατάσταση μεταξύ της αποστολής της διεύθυνσης της slave συσκευής από τον master και της εγγραφής του καταχωρητή αφού απαιτείται η αποστολή της διεύθυνσης του καταχωρητή στον οποίο επιθυμούμε να γίνει εγγραφή.



Σχήμα 4.6: Διαδικασία εγγραφής καταχωρητή R0x09:0—Value 0x0284

Επιπρόσθετα στην διαδικασία ανάγνωσης απαιτείται η αποστολή της διεύθυνσης ανάγνωσης της slave συσκευής μετά την αποστολή και επιβεβαίωση του καταχωρητή τον οποίο επιθυμούμε να αναγνώσουμε.



Σχήμα 4.7: Διαδικασία ανάγνωσης καταχωρητή R0x09:0, επιστρεφόμενη τιμή 0x0284

#### 4.4 Ρολόι

Κάθε master δημιουργεί το δικό του ρολόι στην γραμμή SCL. Μια μετάβαση High to Low στην γραμμή SCL προκαλεί τις συσκευές να μετρήσουν την περίοδο LOW. Όταν το ρολόι της συσκευής γίνει Low κρατά την γραμμή SCL σε αυτή την κατάσταση μέχρι να φθάσουμε στην High κατάσταση του ρολογιού. Σε αυτή υπάρχει και πάλι ένας χρόνος παραμονής ώστε να ολοκληρωθούν οι διαδικασίες στην γραμμή SDA.

#### 4.5 Συμπεράσματα

Το I<sup>2</sup>C πρωτόκολλο είναι αρκετά απλό και προσφέρει πολλά πλεονεκτήματα στον σχεδιαστή και κατασκευαστή. Εφόσον χρησιμοποιούνται μόνο δύο γραμμές είναι κατάλληλο για συστήματα όπου πολλές συσκευές συνδέονται στο ίδιο bus. Με το τρόπο αυτό μειώνεται το κόστος και η πολυπλοκότητα του συστήματος εφόσον νέες συσκευές συνδέονται στο σύστημα. Η ύπαρξη όμως δύο γραμμών δεν αποτελεί πάντα πλεονέκτημα αφού υπάρχει πιθανότητα να προστίθεται πολυπλοκότητα σε κάποιο απλό σύστημα από την ανάγκη χειρισμού των διευθύνσεων και επιβεβαιώσεων.

5

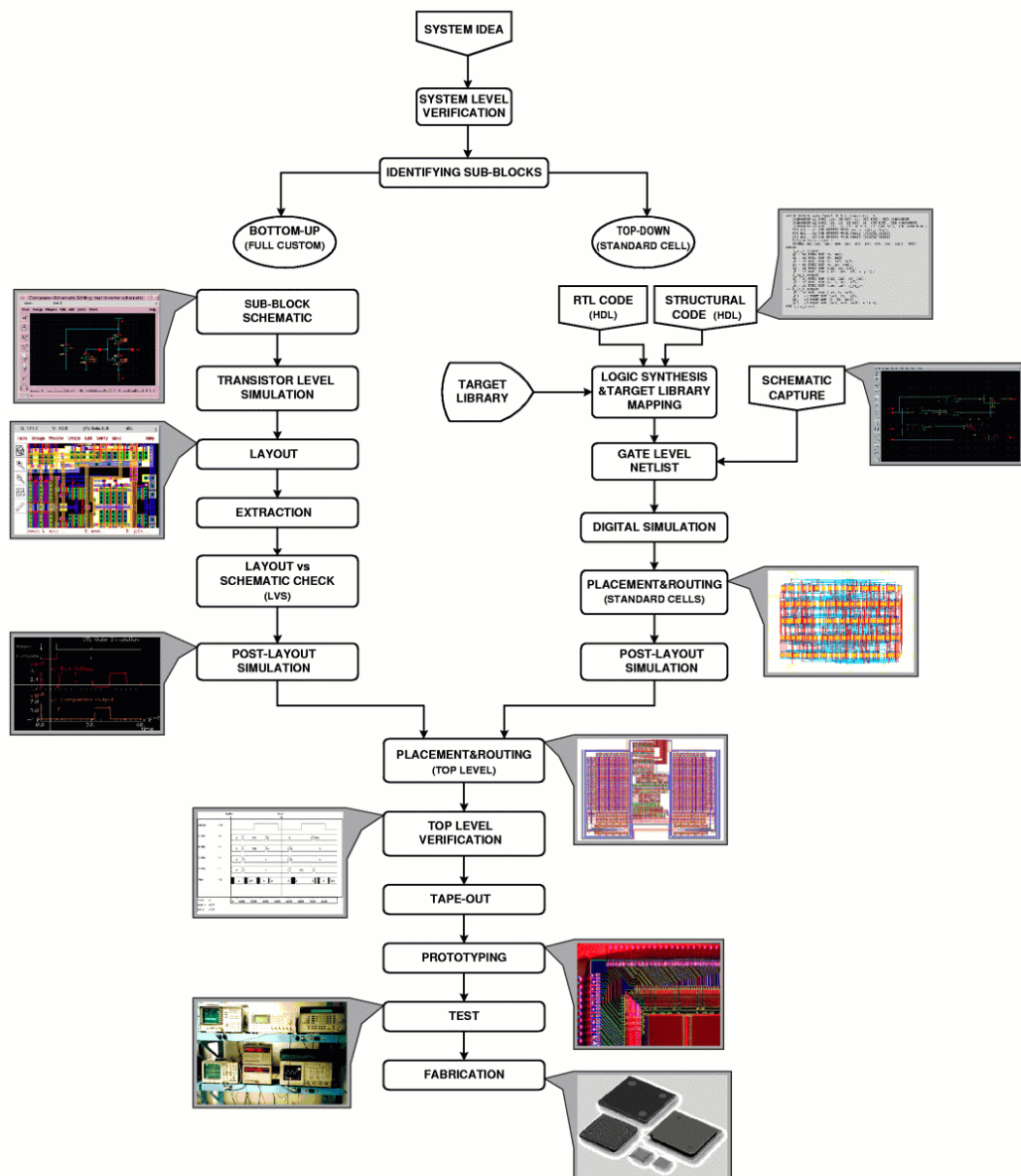
## Ροή πληροφορίας σχεδίασης

Στο παρόν κεφάλαιο περιγράφεται η γενική ροή πληροφορίας σχεδίασης κυκλωμάτων VLSI, τα στάδια δηλαδή που πρέπει να ακολουθηθούν ώστε να ολοκληρωθεί η σχεδίαση ενός κυκλώματος. Επίσης περιγράφεται η τυπική ροή πληροφορίας για την εγκατάσταση κάποιας σχεδίασης σε FPGA. Επιπλέον αναφέρονται κάποια από τα χαρακτηριστικά της γλώσσας περιγραφής υλικού που χρησιμοποιήσαμε (VHDL). Τέλος αναλύονται τα χαρακτηριστικά ενός FPGA ενώ γίνεται αναφορά στο FPGA που είχαμε στη διάθεσή μας.



## 5.1 Γενική ροή πληροφορίας σχεδίασης VLSI

Η γενική ροή της πληροφορίας στη σχεδίαση κυκλωμάτων VLSI απεικονίζεται στο επόμενο σχήμα. Με τον όρο ροή πληροφορίας εννοούμε τα διαφορετικά στάδια που πρέπει να περάσει η πληροφορία ώστε να ολοκληρωθεί η υλοποίηση μιας σχεδίασης.

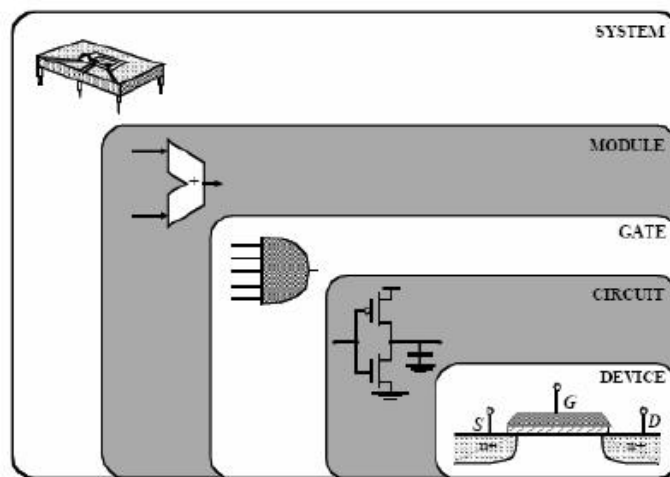


Σχήμα 5.1: Ροή πληροφορίας VLSI [47]

Η μεθοδολογία σχεδίασης bottom-up βασίζεται στην δημιουργία των σχηματικών διαγραμμάτων των στοιχείων του κυκλώματος σε επίπεδο transistor και σε επίπεδο φυσικής σχεδίασης (mask-layout). Εξαιτίας της ευελιξίας που παρέχει η σχεδίαση σε κατώτερα επίπεδα, δίνοντας τη δυνατότητα βελτιστοποίησης του μεγέθους των transistor και της ελαχιστοποίησης των παρασιτικών, η μεθοδολογία αυτή προτιμάται στη σχεδίαση συστημάτων υψηλής απόδοσης, όπως αναλογικών και mixed-signal ενσωματωμένων συστημάτων.

Η μεθοδολογία σχεδίασης top-down βασίζεται κυρίως σε αυτόματη (computer-aided) σύνθεση και δημιουργία του netlist του επιπέδου πυλών, χρησιμοποιώντας μια behavioral ή structural HDL περιγραφή [47]. Η λογική στη συνέχεια γίνεται map σε μια standard-cell βιβλιοθήκη ή σε ένα FPGA. Η μεθοδολογία αυτή ταιριάζει απόλυτα σε ψηφιακά κυκλώματα με σχετικά μικρές χρονικές απαιτήσεις και επιεικής απαιτήσεις από άποψη απόδοσης μεγέθους (area-performance).

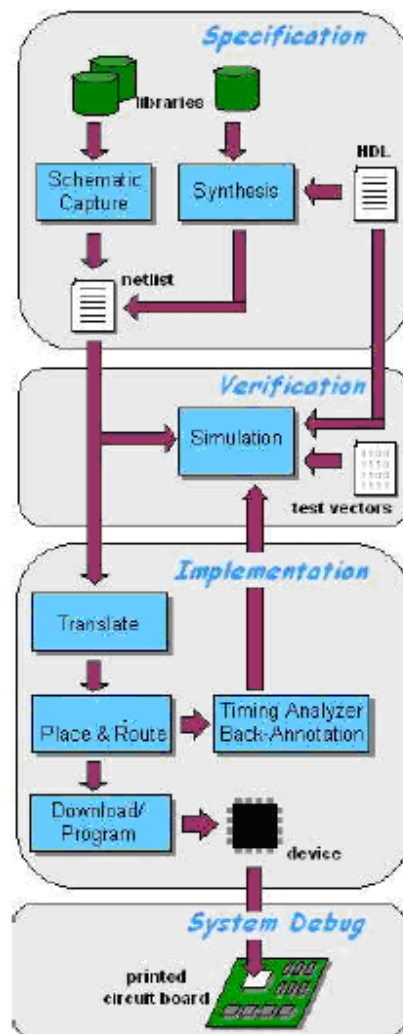
Η διαδικασία από το επίπεδο φυσικής σχεδίασης στη τελική κατασκευή απαιτεί συνήθως πολλές επαναλήψεις για την ικανοποίηση των απαιτήσεων απόδοσης μέσω του σταδίου επαλήθευσης (top level verification), την παραγωγή ενός πρωτοτύπου του chip και τον εκτεταμένο έλεγχο για τον προσδιορισμό και την διόρθωση (debugging) πιθανών σχεδιαστικών λαθών.



Σχήμα 5.2: Σχεδίαση ολοκληρωμένου κυκλώματος

## 5.2 Ροή πληροφορίας σχεδίασης

Στο επόμενο σχήμα απεικονίζεται η ροή πληροφορίας για την εγκατάσταση κάποιας σχεδίασης σε FPGA. Η ολοκλήρωση της σχεδίασης όπως φαίνεται και στο σχήμα απαιτεί την ολοκλήρωση πολλών διαφορετικά σταδίων.



Σχήμα 5.3: Ροή πληροφορίας FPGA [40]

Αρχικά λοιπόν, αφού έχει συλληφθεί η αρχική ιδέα του κυκλώματος, προσδιορίζονται οι απαιτήσεις απόδοσης (Design Specifications), όπως η ταχύτητα καθώς και οι περιορισμοί της σχεδίασης, όπως το μέγεθος και η κατανάλωση ισχύος. Στη συνέχεια γίνεται περιγραφή του κυκλώματος σε μια γλώσσα περιγραφής υλικού (HDL

Description) όπως η VHDL ή η Verilog. Ο κώδικας σε κάθε περίπτωση πρέπει να γίνεται compile για να επιβεβαιώνεται η σωστή σύνταξη. Έπειτα ακολουθεί η λειτουργική προσομοίωση (Presynthesis – Functional Simulation) του κυκλώματος όπου επιβεβαιώνεται η σωστή λειτουργία αλλά και προσδιορίζονται τυχόν λάθη τα οποία και διορθώνονται [37].

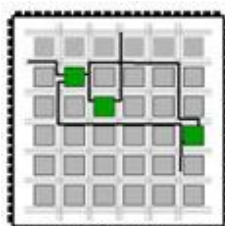
Επόμενο στάδιο αποτελεί η σύνθεση όπου ελέγχεται το κύκλωμα για τυχόν λάθη και μετατρέπεται από αρχείο ‘κειμένου’ (text-based) σε αρχείο NGC netlist. Το αρχείο αυτό είναι μη αναγνώσιμο και περιγράφει την υλοποίηση του κυκλώματος σε πολύ χαμηλό επίπεδο. Σε αυτό το στάδιο έχουμε τη δυνατότητα να δούμε τη σχηματική περιγραφή του κυκλώματος και να φθάσουμε μέχρι και σε επίπεδο πυλών. Σε αυτό το βήμα προκύπτουν κάποιες εκτιμήσεις όσον αφορά το χρόνο και τις καθυστερήσεις εκτέλεσης.

Στη συνέχεια ακολουθούν τα στάδια της υλοποίησης με πρωταρχικό το στάδιο τη μετάφραση (translate) το οποίο ελέγχει τη σχεδίαση και διαβεβαιώνει ότι το netlist είναι σύμφωνο με την επιλεγμένη αρχιτεκτονική. Επιπλέον ελέγχεται το αρχείο περιορισμών (ucf) για οποιαδήποτε αντίφαση. Ουσιαστικά το βήμα αυτό ετοιμάζει τη σχεδίαση για χρήση στο FPGA. Το αρχείο ucf είναι ένα αρχείο περιορισμών το οποίο δημιουργούμε νωρίτερα σε κάποια φάση της σχεδίασης. Οι περιορισμοί αυτοί μπορεί να είναι χωρικοί ή χρονικοί και χρησιμοποιούνται για να θέσουν την απαιτούμενη συχνότητα ή να προσδιορίσουν τα απαιτούμενα pin-out της σχεδίασης.

Το στάδιο MAP κατανέμει τη σχεδίαση στους πόρους του FPGA. Προφανώς εάν η σχεδίαση είναι πολύ μεγάλη για την επιλεγμένη συσκευή τότε το στάδιο αυτό αποτυγχάνει. Στο στάδιο αυτό χρησιμοποιείται το UCF αρχείο για την παρακολούθηση του χρονισμού και υπάρχει ενδεχόμενο πρόσθεσης κι άλλης λογικής με στόχο την ικανοποίηση των χρονικών απαιτήσεων. Το MAP έχει τη δυνατότητα τοποθέτησης της σχεδίασης σε πίνακες ‘look up’ για την δημιουργία της βέλτιστης υλοποίησης. Ουσιαστικά το mapping είναι η διαδικασία της ανάθεσης λογικών στοιχείων της σχεδίασης σε συγκεκριμένα φυσικά στοιχεία τα οποία υλοποιούν λογικές συναρτήσεις σε κάποια συσκευή.

Το στάδιο Place And Route (PAR) ακολουθεί το mapping. Στη φάση Place τα λογικά blocks ανατίθενται σε συγκεκριμένες θέσεις της συσκευής. Στη φάση Routing,

στα λογικά blocks ανατίθενται συγκεκριμένα στοιχεία διασύνδεσης της συσκευής. Εάν έχουν προσδιοριστεί χρονικοί περιορισμοί ο placer μετακινεί τα λογικά blocks πιο κοντά και ο router προσπαθεί να επιλέξει πιο γρήγορη διασύνδεση. Ουσιαστικά επιλέγεται η καλύτερη θέση για κάθε block και η καλύτερη δρομολόγηση με την ελάχιστη καθυστέρηση.



**Σχήμα 5.4:** Place And Route

Σε αυτό το σημείο είναι καλή πρακτική να προσομοιώσουμε ξανά το κύκλωμα εκτελώντας Post Map, Post Translate και Post Place And Route προσομοίωση. Καθώς οι καθυστερήσεις που προστέθηκαν από τα LUTs και τα Flip Flops και οι καθυστερήσεις δρομολόγησης είναι πλέον γνωστές μπορούμε να χρησιμοποιήσουμε την πληροφορία αυτή για να εκτελέσουμε προσομοίωση χρονισμού για να διαπιστωθεί η ικανοποίηση των περιορισμών και να προσδιοριστούν οι πραγματικές καθυστερήσεις.

Τελικά ένα πρόγραμμα που ονομάζεται bitgen παίρνει την έξοδο του Place And Route και δημιουργεί ένα προγραμματισμένο bitstream. Μπορεί όμως να μην είναι απαραίτητη η δημιουργία ενός bit αρχείου σε κάθε υλοποίηση αφού ο σχεδιαστής μπορεί να ενδιαφέρεται μόνο να διαβεβαιώσει ότι κάποιο συγκεκριμένο τμήμα της σχεδίασης περνά με επιτυχία τη προσομοίωση χρονισμού.

### **5.3 Γλώσσα περιγραφής υλικού VHDL**

Το κύκλωμα επιλέχθηκε να περιγραφεί στη γλώσσα VHDL η οποία αποτελεί ακρωνύμιο της VHSIC (Very High Speed Integrated Circuit) [6]. Σε αυτή έχουμε τη δυνατότητα περιγραφής κυκλωμάτων ως προς τη δομή (structure), τη ροή δεδομένων (dataflow) ή τη συμπεριφορά (behavior) τους [5]. Επιπλέον υπάρχει η δυνατότητα περιγραφής της χρονικής συμπεριφοράς (timing) των κυκλωμάτων για εκτέλεση προσομοίωσης.

Γενικά χρησιμοποιείται σε συνδυασμό με έναν μεταφραστή (compiler) για σύνθεση κυκλωμάτων σε ολοκληρωμένα ειδικής σχεδίασης (custom design) ή υλοποίηση σε κάποια διάταξη προγραμματιζόμενης λογικής (PLD ή FPGA), καθώς και από εργαλεία προσομοίωσης λειτουργίας

Κάποιοι από τους λόγους προτίμησής της είναι ότι αποτελεί βιομηχανικό πρότυπο (IEEE 1164 standard - 1993) και ως εκ τούτου είναι ανεξάρτητη του εργαλείου σχεδίασης CAD και του κατασκευαστή ή/και του τρόπου υλοποίησης του κυκλώματος (παρέχει δηλαδή φορητότητα και συμβατότητα σχεδίασης). Επίσης η χρήση προγράμματος (κώδικα) αντί σχηματικών είναι αποτελεσματικότερη για τη σχεδίαση μεγάλων και πολύπλοκων κυκλωμάτων, τόσο λόγω ταχύτερης σύλληψης (capture) όσο και ευκολότερης διαχείρισης και τροποποίησης της σχεδίασης. Επιπλέον έχουμε τη δυνατότητα επαναχρησιμοποίησης σχεδιάσεων και πρόσβασης σε έτοιμες βιβλιοθήκες τρίτων [5], [7].

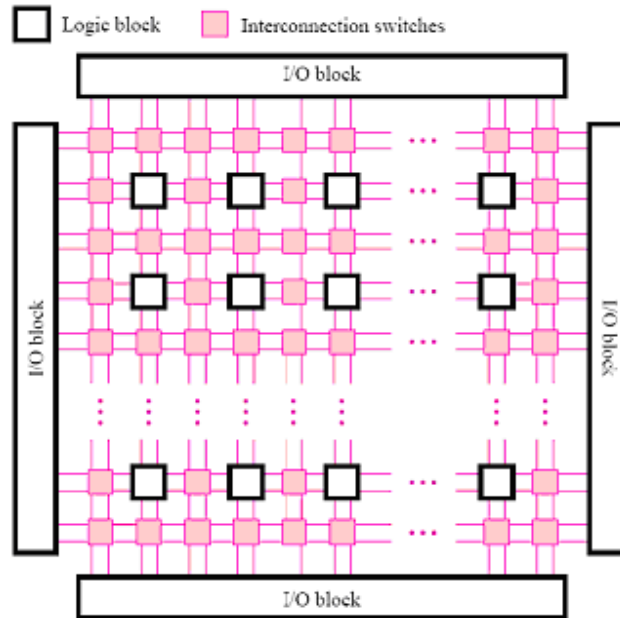
Τέλος η περιγραφή συμπεριφοράς (εκτός από δομή) επιτρέπει τη σχεδίαση σε υψηλότερα επίπεδα αφαίρεσης, όπου ο μεταφραστής δημιουργεί αυτόματα το κύκλωμα με τη χρήση έτοιμων βιβλιοθηκών και τεχνικών σύνθεσης. Εξάλλου στην πράξη η ανάπτυξη ψηφιακών κυκλωμάτων γίνεται σχεδόν αποκλειστικά σε γλώσσες περιγραφής υλικού, με κυριότερη εξ αυτών τη VHDL.

Αναφέρουμε επίσης ότι ένα από τα μειονεκτήματά της είναι η εμφάνιση μνημονικής συμπεριφοράς (ουσιαστικά ενός βρόχου ανάδρασης) η οποία είναι ανεπιθύμητη για τα συνδυαστικά κυκλώματα αλλά αποτελεί τη βάση για την ανάπτυξη των ακολουθιακών κυκλωμάτων

#### **5.4 Διάταξη προγραμματιζόμενης λογικής - FPGA**

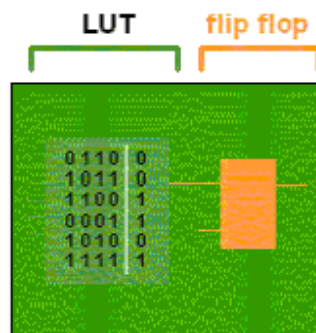
Ένα FPGA χαρακτηρίζεται σαν ψηφιακό λογικά προγραμματιζόμενο chip (programmable digital logic chip) [37]. Είναι δισδιάστατες διατάξεις (προγραμματιζόμενων) λογικών βαθμίδων οι οποίες διασυνδέονται μεταξύ τους μέσω καλωδίων και (προγραμματιζόμενων) λογικών διακοπών τα οποία βρίσκονται υπό τον πλήρη έλεγχο του σχεδιαστή. Αυτό σημαίνει ότι ο χρήστης μπορεί να σχεδιάσει, να

προγραμματίζει και να κάνει αλλαγές στο κύκλωμα όποτε το επιθυμεί. Παρακάτω φαίνεται η γενική δομή μιας διάταξης FPGA.



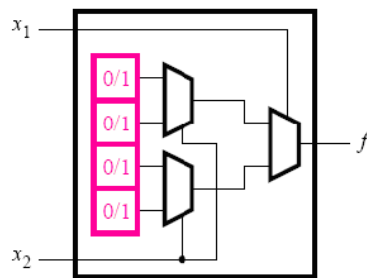
Σχήμα 5.5: Γενική δομή μιας διάταξης FPGA

Υπάρχουν 2 είδη FPGAs. Τα επαναπρογραμματιζόμενα και αυτά τα οποία προγραμματίζονται μια μόνο φορά. Τα επαναπρογραμματιζόμενα βασίζονται σε λογικά κελιά SRAM τα οποία αντί για τις συμβατικές πύλες χρησιμοποιούν τα λεγόμενα Look Up Tables (LUT) που προσδιορίζουν την έξοδο με βάση τις τιμές των εισόδων. Τα ψηφία των SRAM χρησιμοποιούνται και για την δημιουργία των συνδέσεων. Στο επόμενο σχήμα φαίνονται οι διαφορετικοί συνδυασμοί των 4 εισόδων που καθορίζουν την έξοδο.



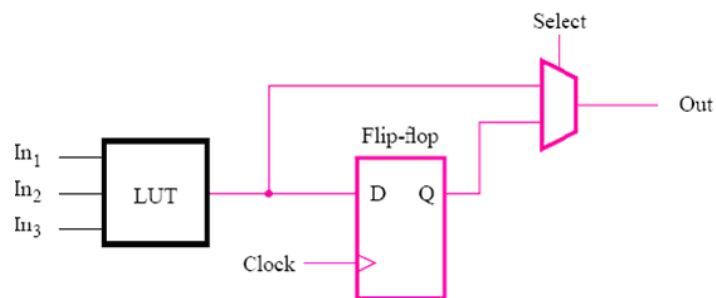
Σχήμα 5.6: Λογική βαθμίδα επαναπρογραμματιζόμενων FPGA

Οι λογικές βαθμίδες λοιπόν είναι τύπου lookup table (LUT), με στοιχεία - latches - αποθήκευσης τιμών (0 και 1) και κυκλώματα επιλογής για την υλοποίηση λογικών συναρτήσεων. Η υλοποίηση κάθε λογικής συνάρτησης απαιτεί την αποθήκευση των αντίστοιχων τιμών από τον πίνακα αλήθειας της στα latches. Κάθε λογική βαθμίδα εκτελεί μια μικρή λειτουργία και καθώς ενώνεται με άλλες είναι σε θέση να υλοποιήσουν μια πολύπλοκη λειτουργία.



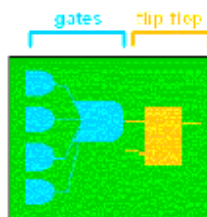
Σχήμα 5.7: Look Up Table

Επιπλέον σε κάθε βαθμίδα υπάρχουν flip-flops για την υλοποίηση ακολουθιακής λογικής



Σχήμα 5.8: LUT και flip flops

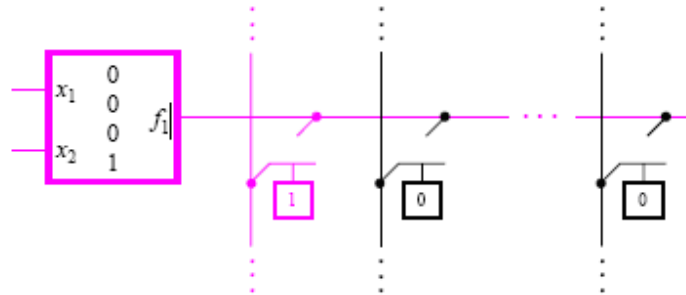
Το άλλο είδος των FPGA προγραμματίζεται μόνο μια φορά και η λογική που χρησιμοποιεί βασίζεται στις συμβατικές πύλες και χαρακτηρίζεται σαν One-Time Programmable (OTP).



Σχήμα 5.9: Λογική βαθμίδα OTP (One Time Programmable) FPGA

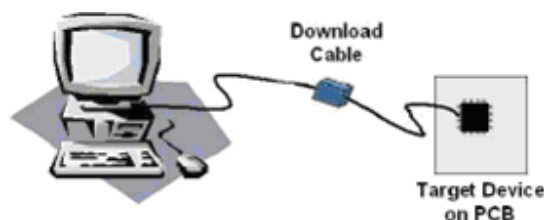


Όσον αφορά τους προγραμματιζόμενους διακόπτες, αυτοί χαρακτηρίζονται ως ανοικτοί ή κλειστοί όταν εφαρμόζεται αντίστοιχα 0 ή 1 σε κάποιο κατάλληλο ακροδέκτη που τους ελέγχει. Η πληροφορία ενεργοποίησης των διακοπών προγραμματίζεται, όπως και στις λογικές βαθμίδες, σε κατάλληλα στοιχεία αποθήκευσης ή latches.



Σχήμα 5.10: Διακόπτες FPGA

Γενικά για την υλοποίηση μιας σχεδίασης αρχικά γίνεται περιγραφή της επιθυμητής λογικής συνάρτησης, "logic function", είτε μέσω της δημιουργίας μιας σχηματικής περιγραφής, είτε μέσω ενός αρχείου κειμένου. Στη συνέχεια γίνεται προσομοίωση της λογικής συνάρτησης "logic function" χρησιμοποιώντας κάποιο λογισμικό που παρέχεται από τον δημιουργό του FPGA. Αυτό δημιουργεί ένα binary αρχείο (.bit) που μπορεί να εγκατασταθεί στο FPGA μέσω ενός καλωδίου από τον υπολογιστή.



Σχήμα 5.11: Εγκατάσταση σχεδίασης

Έτσι το FPGA συμπεριφέρεται σύμφωνα με την λογική συνάρτηση. Κάποιος μπορεί να εγκαταστήσει κυκλώματα σε FPGA όσες φορές επιθυμεί χωρίς να υπάρχει κάποιο όριο. Αν γίνει κάποιο λάθος στη σχεδίαση απλά διορθώνεται η λογική συνάρτηση κάνουμε re-compile και εγκαθιστούμε ξανά τη σχεδίαση. Σημειώνουμε ότι τα FPGA χάνουν την λειτουργικότητά τους (σβήνουν την προγραμματιζόμενη συνάρτηση) όταν αποσυνδέουμε την τροφοδοσία.

Γενικά τα FPGA απαιτούν κάποια τάση, για να λειτουργήσουν τα flip-flop και οι λογικές πύλες. Σε αυτά υπάρχουν 2 είδη pins, τα αφοσιωμένα (dedicated pins) και τα pins του χρήστη. Φτιάχνονται από ένα βασικό λογικό κελί, "logic-cell", το οποίο διπλασιάζεται 100άδες ή και 1000άδες φορές.

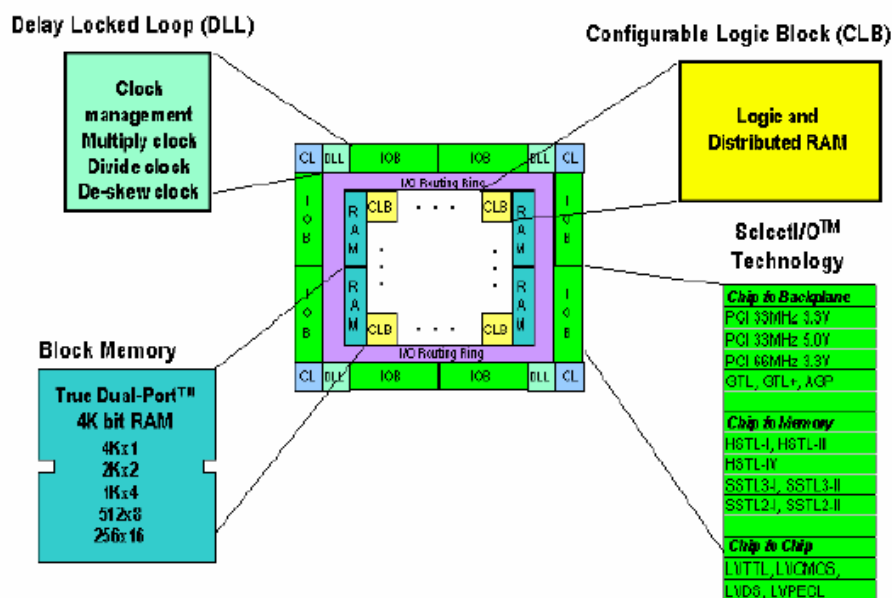
Συνήθως μια σχεδίαση FPGA είναι σύγχρονη και κάθε θετικός παλμός επιτρέπει στο D-flip flop να αλλάξει κατάσταση. Ένα ρολόι μπορεί να οδηγήσει πολλές εισόδους ταυτόχρονα.

## 5.5 FPGA σχεδίασης



Σχήμα 5.12: Spartan IIE

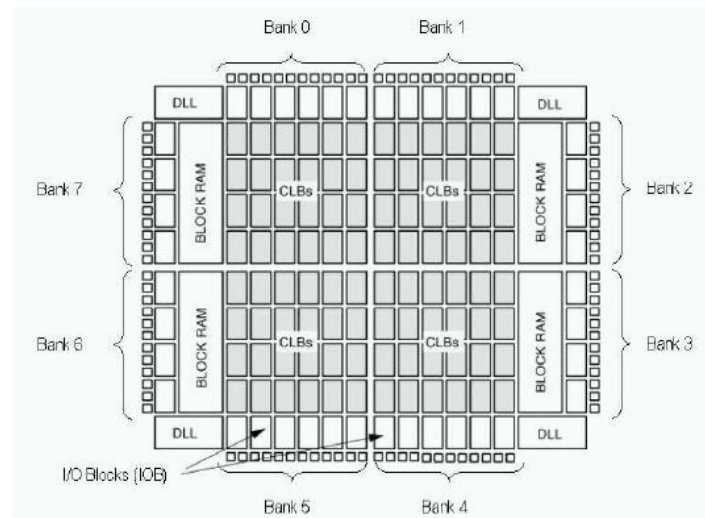
Το FPGA που είχαμε στη διάθεσή μας και για το οποίο υλοποιήθηκαν τα κυκλώματα ήταν το SPARTAN IIE της εταιρείας Xilinx [43] και η αρχιτεκτονική του απεικονίζεται στο επόμενο σχήμα.



Σχήμα 5.13: Αρχιτεκτονική Spartan IIE [37]

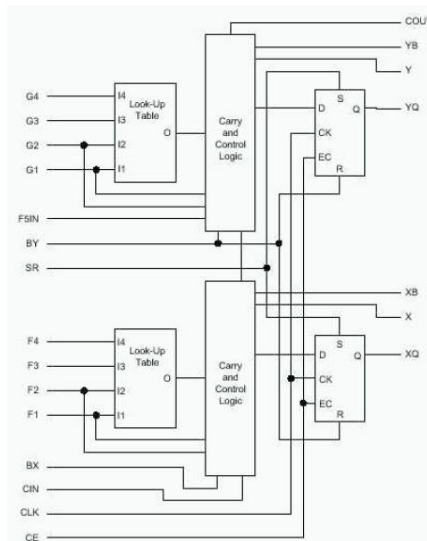
Η βασική δομή CLB περιλαμβάνει κατανεμημένη RAM και εκτελεί βασικές λειτουργίες. Τα τέσσερα DLLs χρησιμοποιούνται για την κατανομή του ρολογιού και μπορούν να χρησιμοποιηθούν για πολλαπλασιασμό και διαίρεση του ρολογιού.

Στο επόμενο σχήμα φαίνονται, περιφερειακά του FPGA, οι ομάδες των εισόδων – εξόδων (Banks), τα κυκλώματα ελέγχου ρολογιού, οι βασικές λογικές βαθμίδες και τα κυκλώματα RAM.



**Σχήμα 5.14:** Block διάγραμμα Spartan III

Το βασικό δομικό στοιχείο του CLB είναι το λογικό κελί (LC), το οποίο περιέχει μια γεννήτρια 4-εισόδων η έξοδος της οποίας οδηγεί την έξοδο του CLB και την είσοδο του D flip flop, λογική μεταφοράς και ελέγχου και ένα στοιχείο αποθήκευσης. Κάθε CLB περιέχει 4 λογικά κελιά οργανωμένα σε 2 όμοια τμήματα (slices). Οι γεννήτριες συναρτήσεων (function generators) υλοποιούνται σαν Look Up Tables (LUT) 4 εισόδων. Επιπρόσθετα κάθε LUT μπορεί να παρέχει μια σύγχρονη RAM 16 x 1 ψηφίων. Οι δύο LUT σε ένα τμήμα μπορούν να συνδυαστούν και να δώσουν μια RAM 16x2 ή 32x1 ψηφίων. Τα στοιχεία αποθήκευσης μπορούν να διαμορφωθούν σαν D flip flops ή latches. Επίσης χρησιμοποιούνται πολυπλέκτες για την δρομολόγηση των σημάτων.



Σχήμα 5.15: Τμήμα CLB, 2 λογικά κελιά

Το DLL (delay locked loop) σχετίζεται με κάθε γενικό ρολόι εισόδου και επιτυγχάνει να απαλείψει το skew μεταξύ του ρολογιού εισόδου και των εσωτερικών ρολογιών επιτυγχάνοντας συγχρονισμό των ακμών του ρολογιού που φθάνουν εσωτερικά στα flip flop με τις ακμές του ρολογιού στην είσοδο.

Γενικά ένα FPGA τοποθετείται σε ένα board το οποίο εκτός των άλλων περιέχει και άλλα κυκλώματα, όπως μνήμες ακόμα και επεξεργαστές για DSP εφαρμογές και έχει τη δυνατότητα σύνδεσης σε περιφερειακές συσκευές [42] όπως οθόνες LCD και ελεγκτές ή και άλλα board που χρησιμοποιούνται για να δίνουμε τις εισόδους και να ελέγχουμε τις εξόδους. Ένα τέτοιο board όπου διακρίνεται το FPGA στο κέντρο του, απεικονίζεται στο επόμενο σχήμα.



Σχήμα 5.16: Board με FPGA [43]

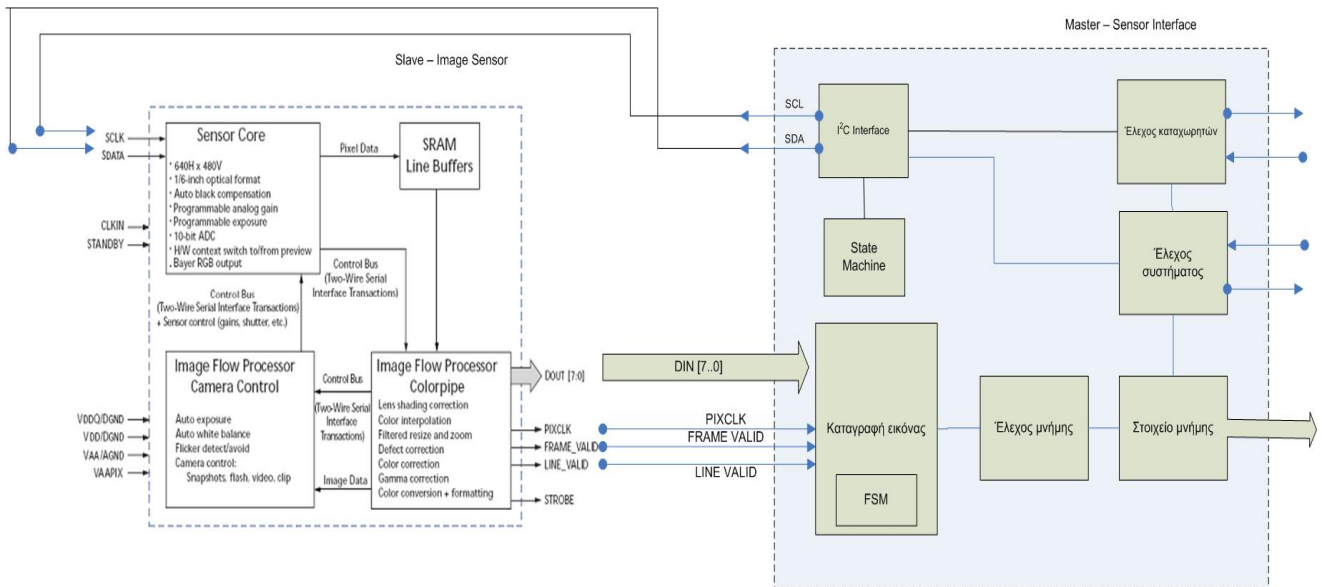
6

## Αρχιτεκτονική Συστήματος

Στο παρόν κεφάλαιο περιγράφεται η αρχιτεκτονική του συστήματος διεπαφής που προτείνουμε. Σε κάθε επιμέρους ενότητα περιγράφεται η αρχιτεκτονική κάθε κυκλώματος απαραίτητο για την υλοποίηση του συστήματος. Επιπλέον σε κάποιες περιπτώσεις προτείνονται εναλλακτικές αρχιτεκτονικές σε μορφή μηχανών πεπερασμένων καταστάσεων όπως για το πρωτόκολλο επικοινωνίας του αισθητήρα.

## 6.1 Εισαγωγή

Η διεπαφή μεταξύ του αισθητήρα και του συστήματος απεικονίζεται στο επόμενο σχήμα όπου διακρίνονται τα επιμέρους κυκλώματα και η αρχιτεκτονική του αισθητήρα. Το σύστημα λοιπόν της διεπαφής αποτελείται από το κύκλωμα που υλοποιεί το πρωτόκολλο επικοινωνίας και το κύκλωμα καταγραφής εικόνας. Επιπλέον ένα κύκλωμα ελέγχου της μνήμης ελέγχει την εγγραφή και ανάγνωση στη μνήμη. Επίσης προτείνονται κυκλώματα έλεγχου του συστήματος και των καταχωρητών.

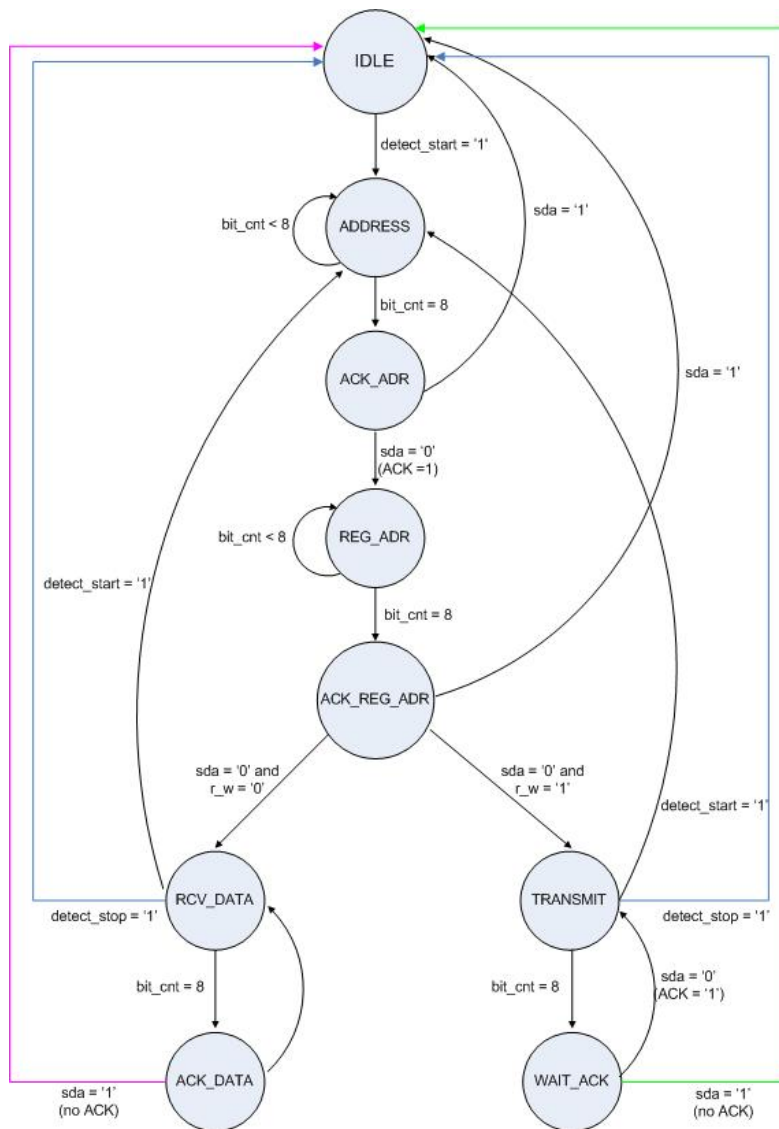


Σχήμα 6.1: Αρχιτεκτονική συστήματος

## 6.2 Πρωτόκολλο επικοινωνίας I<sup>2</sup>C

Το I<sup>2</sup>C είναι ένα πρωτόκολλο αμφίδρομης επικοινωνίας το οποίο χρησιμοποιείται για την ανταλλαγή δεδομένων μεταξύ των συσκευών. Χρησιμοποιείται στο συγκεκριμένο σύστημα για τον προγραμματισμό των κατάλληλων καταχωρητών του αισθητήρα και για την ανάγνωση των τιμών κάποιων καταχωρητών. Η μηχανή πεπερασμένων καταστάσεων η οποία ελέγχει την λειτουργία του πρωτοκόλλου φαίνεται στο επόμενο σχήμα. Η μηχανή αυτή ελέγχει έναν μετρητή, ο οποίος μετράει κάθε φορά τον αριθμό των bits που μεταφέρθηκαν και δύο ολισθητές, ο ένας μεταφέρει την

διεύθυνση του αισθητήρα και την διεύθυνση του καταχωρητή, ενώ ο άλλος τα δεδομένα που πρόκειται να παραληφθούν ή να αποσταλούν. Παρατηρούμε ότι η επόμενη μηχανή πεπερασμένων καταστάσεων είναι βασισμένη στο πρωτόκολλο I<sup>2</sup>C με προσθήκη όμως της κατάστασης αποστολής της διεύθυνσης του καταχωρητή. Επιπλέον στην κατάσταση ADDRESS στέλνουμε είτε την διεύθυνση ανάγνωσης είτε τη διεύθυνση εγγραφής της slave συσκευής και ανάλογα με το τελευταίο bit, δίνουμε τιμή στο σήμα r\_w και στη κατάσταση ACK\_REG\_ADR μπορούμε να επιλέξουμε την κατάσταση στην οποία θα μεταβούμε. Το r\_w γίνεται ίσο με τη μονάδα όταν πρόκειται να γίνει εγγραφή και ίσο με το μηδέν για ανάγνωση.



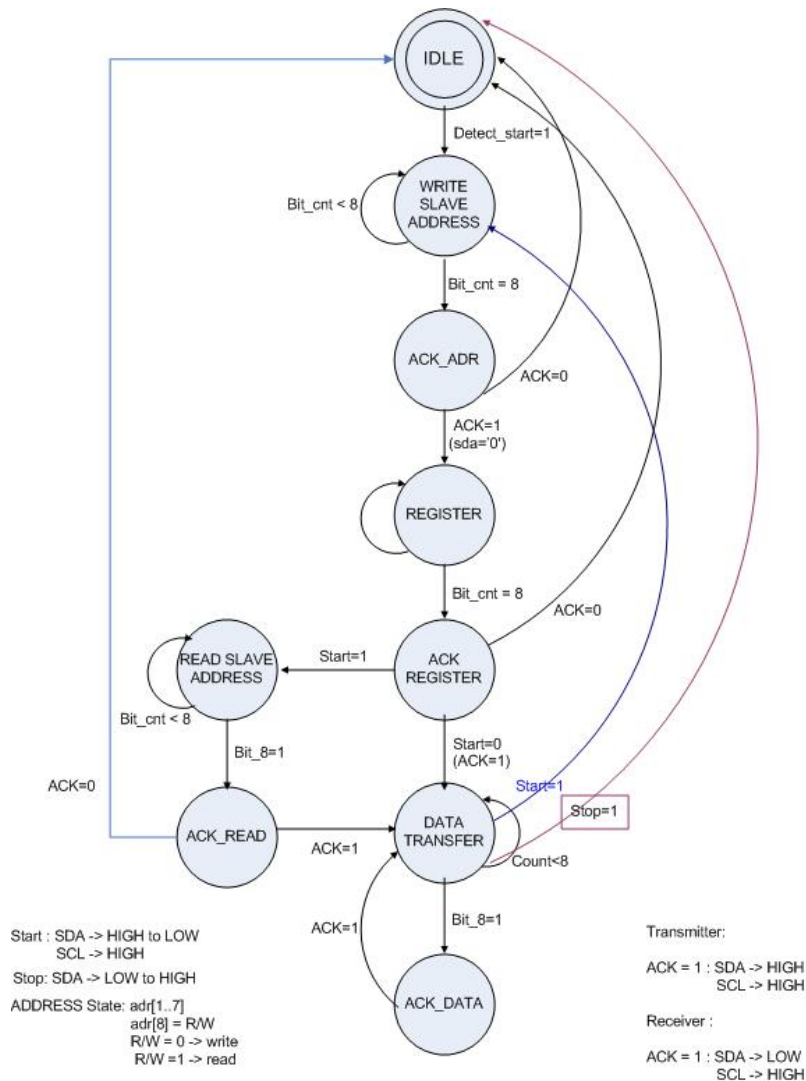
Σχήμα 6.2: I<sup>2</sup>C Πρωτόκολλο επικοινωνίας

Όνομα κατάστασης	Λειτουργία
IDLE	Κατάσταση αδράνειας, στην οποία δεν συμβαίνει τίποτα, τα bus είναι άδεια (SDA και SCL HIGH) και περιμένουμε να στείλει ο master κάποιο σήμα έναρξης επικοινωνίας (start =1) με κάποια slave συσκευή (image sensor) θέτοντας το SDA: Low και διατηρώντας το SCL: High.
ADDRESS	Αναμονή αποστολής της διεύθυνσης της slave συσκευής από τον master. Τα πρώτα 7 bit (address) αποτελούν τη διεύθυνση της slave συσκευής ενώ το 8 <sup>ο</sup> bit (R/W) αν είναι 0 σημαίνει ότι θα γίνει εγγραφή (master transmit) του master στον slave ενώ αν είναι 1 σημαίνει ότι θα ακολουθήσει λήψη δεδομένων (master receive) από τον slave.
ACK_ADDRESS	Αναμονή λήψης σήματος επιβεβαίωσης από τον αισθητήρα.
REG_ADR	Αναμονή αποστολής της διεύθυνσης (8-bit) του καταχωρητή του οποίου πρόκειται να γίνει εγγραφή ή ανάγνωση.
ACK_REG_ADR	Αναμονή λήψης σήματος επιβεβαίωσης από τον αισθητήρα.
RCV_DATA	Αναμονή λήψης δεδομένων από τον καταχωρητή.
ACK_DATA	Αποστολή σήματος επιβεβαίωσης από το σύστημα (tack)
TRANSMIT	Αναμονή αποστολής δεδομένων προς τον αισθητήρα για τον προγραμματισμό των καταχωρητών.
WAIT_ACK	Αναμονή λήψης σήματος επιβεβαίωσης από τον αισθητήρα.

**Πίνακας 6.1:** Περιγραφή καταστάσεων του I<sup>2</sup>C Πρωτοκόλλου επικοινωνίας

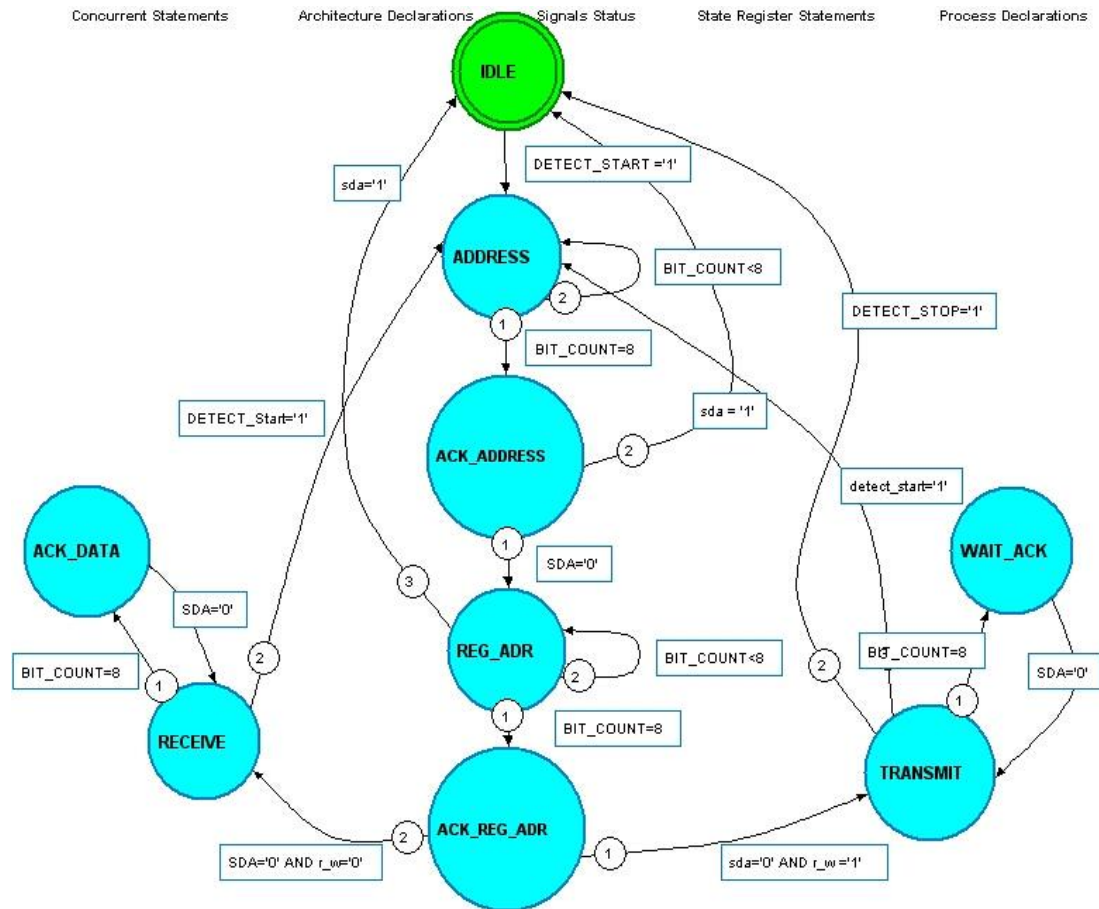
Ο αισθητήρας που περιγράψαμε στο τρίτο κεφάλαιο περιγράφει ένα ελαφρώς παραλλαγμένο πρωτόκολλο στην περίπτωση κυρίως της ανάγνωσης. Η διαδικασία της ανάγνωσης λοιπόν γίνεται ακολουθώντας την ίδια διαδικασία μέχρι και την αποστολή και επιβεβαίωση της διεύθυνσης του καταχωρητή οπότε και ο master, είτε στέλνει σήμα start και τη διεύθυνση ανάγνωσης της slave συσκευής, είτε δεν στέλνει start οπότε και μεταβαίνει στην κατάσταση μεταφοράς. Αν στείλει start και επιβεβαιωθεί η διεύθυνση τότε μεταβαίνει στην κατάσταση μεταφοράς όπου ο master διαβάζει κάποιον καταχωρητή του αισθητήρα. Στο επόμενο σχήμα απεικονίζεται η ελαφρώς παραλλαγμένη μηχανή πεπερασμένων καταστάσεων.





Σχήμα 6.3: I<sup>2</sup>C Πρωτόκολλο επικοινωνίας με πρόσθετα χαρακτηριστικά

Στο σχήμα 6.4 απεικονίζεται η παραπάνω μηχανή πεπερασμένων καταστάσεων σχεδιασμένη στο HDL DESIGNER. Σε αυτό επιλέγοντας τη δημιουργία κώδικα vhdl, συμπεραίνουμε ότι ο παραγόμενος κώδικας είναι όμοιος με αυτόν που υλοποιήσαμε και πολύ καλά οργανωμένος με μικρές διαφορές, οι οποίες πολύ εύκολα διορθώνονται. Μια από αυτές είναι ότι ο κώδικας που περιγράφει τη μηχανή δεν έχει τις καταστάσεις με τη σειρά και δεν ξεκινά από την κατάσταση αδράνειας, κάτι το οποίο πολύ εύκολα μεταβάλλεται.

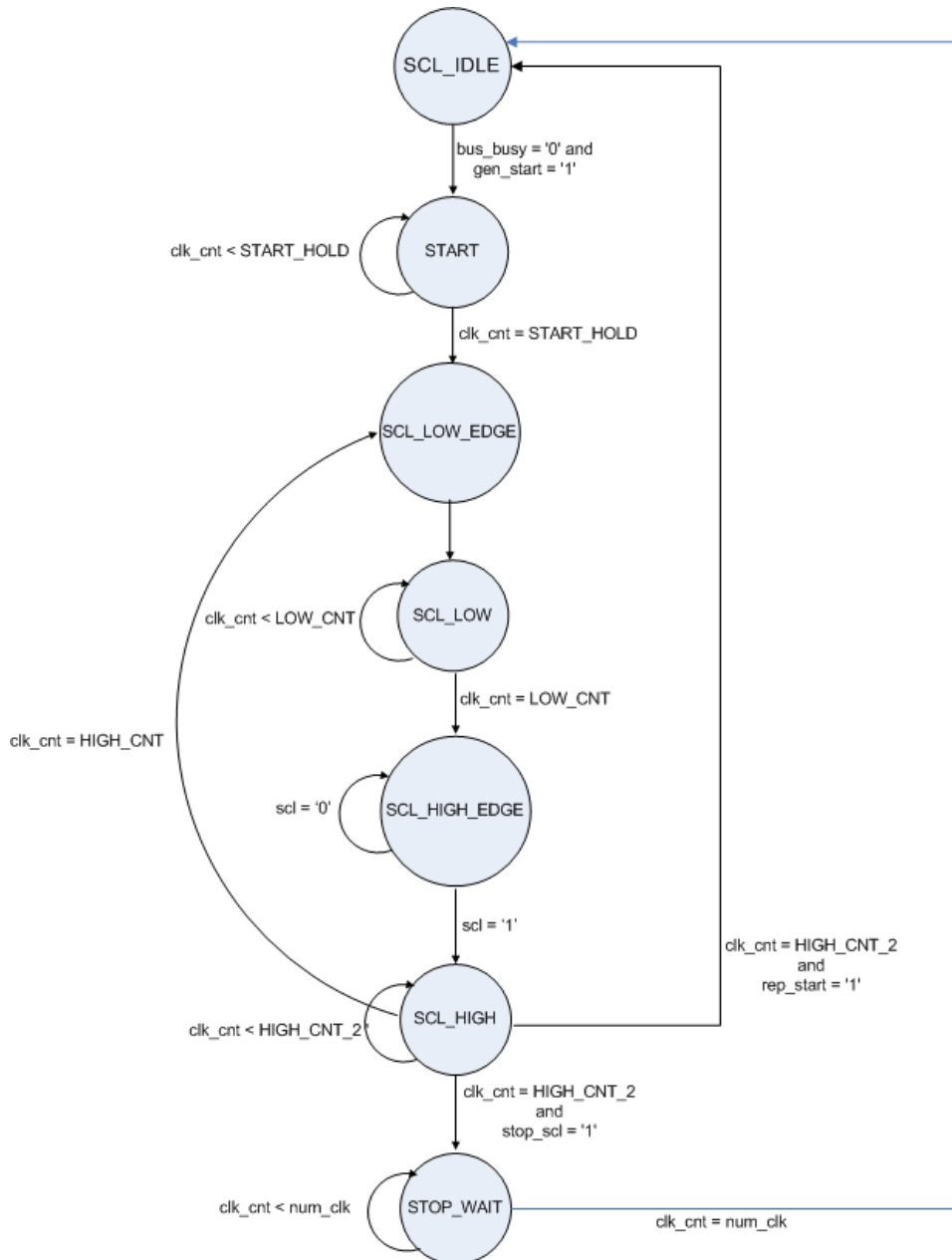


Σχήμα 6.4: Μηχανή σχεδιασμένη στο HDL Designer

### 6.2.1 Παραγωγή scl, sda start και stop σημάτων

Για την λειτουργία του πρωτοκόλλου, είναι απαραίτητη η παραγωγή των σημάτων scl, sda, start και stop όταν βρισκόμαστε σε κατάσταση master. Συνήθως η συχνότητα του ρολογιού είναι περίπου 100 KHz και παράγεται διαιρώντας το ρολόι του συστήματος. Ο αριθμός των κύκλων του ρολογιού που απαιτείται για την παραγωγή ενός ρολογιού 100 kHz προσδιορίζεται από τη μεταβλητή HIGH\_CNT και LOW\_CNT . Επιπλέον καθορίζονται και οι χρόνοι αναμονής στις καταστάσεις έναρξης και μεταφοράς δεδομένων. Μέσω ενός μετρητή μετριούνται οι περιόδοι του ρολογιού που απαιτούνται για την ικανοποίηση των χρονικών απαιτήσεων. Η διαδικασία παραγωγής των σημάτων περιγράφεται από την επόμενη μηχανή πεπερασμένων καταστάσεων.

Η έξοδος του sda από αυτή τη σχεδίαση είναι είτε το σήμα sda που παράγεται από την παρακάτω μηχανή στην περίπτωση start και stop σημάτων ή τα δεδομένα μεταφοράς εφόσον βρισκόμαστε σε master mode.



Σχήμα 6.5: Μηχανή πεπερασμένων καταστάσεων παραγωγής scl και sda σημάτων

Ο παρακάτω πίνακας περιγράφει τη λειτουργικότητα της κάθε κατάστασης της μηχανής πεπερασμένων καταστάσεων του σχήματος 5.

Όνομα κατάστασης	Λειτουργία
SCL_IDLE	Κατάσταση αδράνειας, στην οποία τα bus είναι άδεια. Αναμονή για αίτηση έναρξης.
START	Το scl τίθεται high και το sda low για την παραγωγή σήματος έναρξης και ο μετρητής του ρολογιού ενεργοποιείται. Αναμονή μέχρι να ικανοποιηθεί ο χρονικός περιορισμός START_HOLD.
SCL_LOW_EDGE	Παραγωγή falling edge και μηδενισμός του ρολογιού για μέτρηση της χρονικής περιόδου LOW.
SCL_LOW	Το ρολόι διατηρείται LOW και ο μετρητής μετράει για την ικανοποίηση του LOW χρονικού περιορισμού.
SCL_HIGH_EDGE	Παραγωγή rising edge του ρολογιού
SCL_HIGH	Το ρολόι διατηρείται HIGH και ο μετρητής μετράει για την ικανοποίηση του HIGH χρονικού περιορισμού.
STOP_WAIT	Ο μετρητής μετράει για την ικανοποίηση του χρονικού διαστήματος που μεσολαβεί μεταξύ της έναρξης και λήξης μιας μεταφοράς.

**Πίνακας 6.2:** Περιγραφή καταστάσεων της μηχανής παραγωγής scl και sda σημάτων

Στη συνέχεια ακολουθούν κάποιες παρατηρήσεις σχετικά με το πρωτόκολλο επικοινωνίας.

- Τα δεδομένα στο SDA αλλάζουν μόνο όταν το SCL είναι LOW.
- Το bus είναι ελεύθερο όταν SDA και SCL HIGH
- Δεν χρειάζεται η υλοποίηση της arbitration λειτουργίας που υποστηρίζει το I<sup>2</sup>C πρωτόκολλο αφού στο συγκεκριμένο σύστημα υπάρχει ένας master
- Η slave διεύθυνση του image sensor είναι
  - διεύθυνση εγγραφής : 0XBA : 10111010
  - διεύθυνση ανάγνωσης: 0XBB : 10111011

### 6.3 Καταγραφή εικόνας

Η αρχικοποίηση των καταχωρητών, όπως προαναφέρθηκε, γίνεται μέσω του πρωτοκόλλου επικοινωνίας I<sup>2</sup>C. Εφόσον αρχικοποιηθούν οι κατάλληλοι καταχωρητές του αισθητήρα, όπως αυτοί που είναι υπεύθυνοι για τον προσδιορισμό του μεγέθους της εικόνας, του τύπου εξόδου της εικόνας και οι υπόλοιποι που ρυθμίζουν διάφορες απαραίτητες λειτουργίες του αισθητήρα, είναι δυνατή η σύλληψη ενός πλαισίου- εικόνας (frame) και η αποθήκευση του στη μνήμη.

Η έξοδος του αισθητήρα μπορεί να έχει διάφορες μορφές όπως 4:2:2 YCrCb (CCIR656) ή 4:4:4 565RGB, η οποία είναι η πιο διαδεδομένη και η οποία υιοθετήθηκε στην υλοποίηση του συστήματος. Ο συγκεκριμένος αισθητήρας παράγει πρώτα τα υψηλότερα 8-bit και στη συνέχεια τα κατώτερα 8-bit κάθε pixel και με βάση αυτό το δεδομένο υλοποιήθηκε το συγκεκριμένο σύστημα. Επειδή όμως η έξοδος ενός αισθητήρα μπορεί να ποικίλει λήφθηκε υπόψη και η περίπτωση να λαμβάνονται πρώτα τα κατώτερα 8-bit και στη συνέχεια τα υψηλότερα 8-bit .

Το κύκλωμα αυτό δέχεται σαν είσοδο 8-bit κάθε φορά και σχηματίζει ένα pixel 16-bit τύπου RGB (565). Επιπλέον δέχεται τα σήματα ελέγχου line\_valid και frame\_valid τα οποία προσδιορίζουν την ύπαρξη έγκυρης γραμμής και έγκυρου πλαισίου αντίστοιχα. Τέλος έχει σαν είσοδο το ρολόι pixelclk του αισθητήρα το οποίο συγχρονίζει την μεταφορά των δεδομένων.

Η λειτουργία του κυκλώματος που είναι υπεύθυνο για την σύλληψη της εικόνας και την αποστολή των pixel στη μνήμη, περιγράφεται από την επόμενη μηχανή πεπερασμένων καταστάσεων. Στο σχήμα 6α απεικονίζεται η μηχανή πεπερασμένων καταστάσεων στην περίπτωση που λαμβάνονται πρώτα τα υψηλότερα 8-bit και ακολουθούν τα κατώτερα 8-bit κάθε pixel και στο σχήμα 6β η αντίθετη περίπτωση. Παρατηρούμε ότι οι 2 περιπτώσεις υλοποιούνται χωρίς πολλές διαφορές, τόσο από άποψη αρχιτεκτονικής όσο και σε επίπεδο περιγραφής σε γλώσσα υλικού.



Σχήμα 6.6: Μηχανή πεπερασμένων καταστάσεων καταγραφής εικόνας

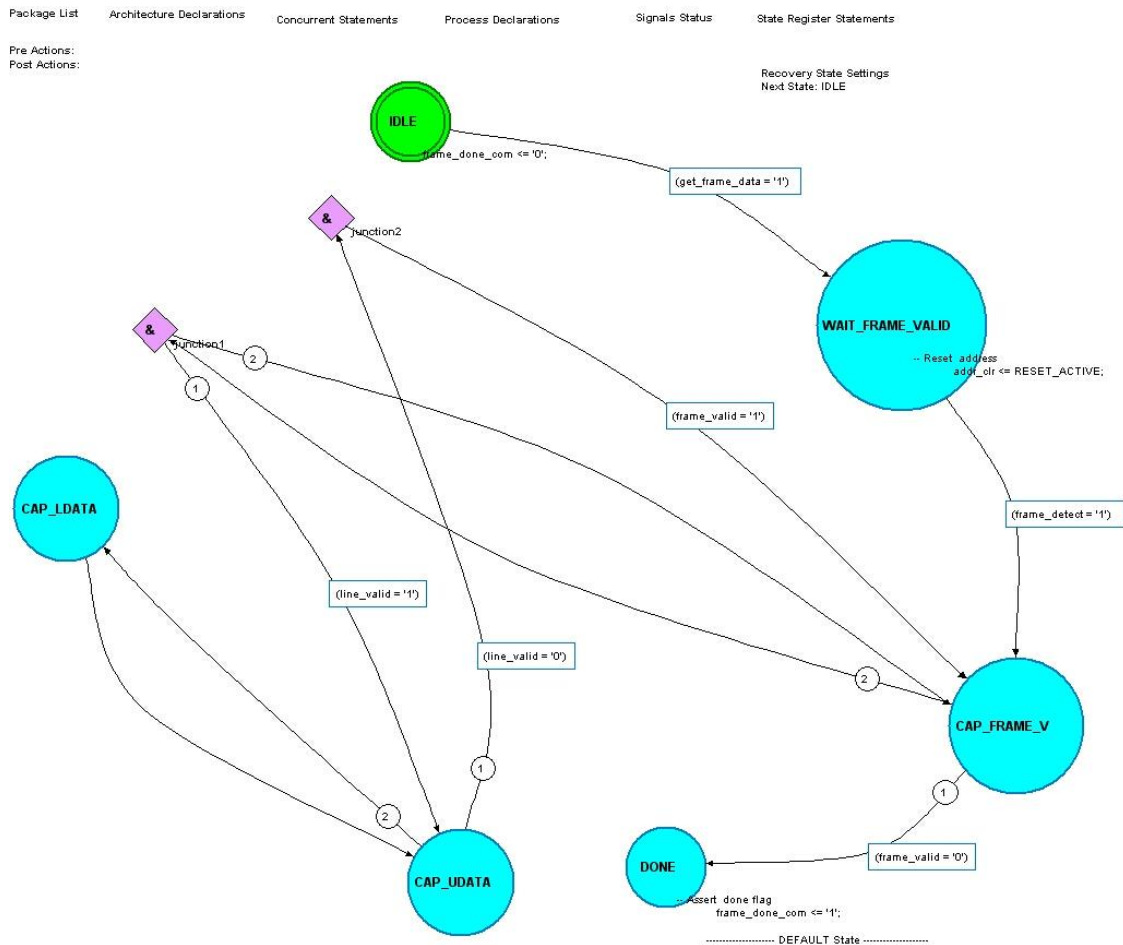
Ο παρακάτω πίνακας περιγράφει τη λειτουργικότητα της κάθε κατάστασης της μηχανής πεπερασμένων καταστάσεων του σχήματος 6α.

Όνομα κατάστασης	Λειτουργία
IDLE	Αναμονή ενεργοποίησης του σήματος ελέγχου get_frame_data.
WAIT_FRAME_VALID	Αναμονή ανίχνευσης του σήματος frame_detect το οποίο σηματοδοτεί την έναρξη λήψης ενός πλαισίου και αρχικοποίηση των διευθύνσεων της μνήμης στο μηδέν.
CAP_FRAME_V	Αναμονή για να τεθεί το σήμα ελέγχου του αισθητήρα line_valid στη μονάδα εφόσον το σήμα frame_valid είναι ενεργοποιημένο, δεν έχει τελειώσει δηλαδή η λήψη της εικόνας.
CAP_UDATA	Λήψη των υψηλότερων τιμών (15 ..8 ) ενός pixel. Ενεργοποίηση των σημάτων ελέγχου της μνήμης wen και addr_cnt_en τα οποία σηματοδοτούν την εγγραφή της μνήμης και την αύξηση κατά μια θέση κάθε φορά της διεύθυνσής της.
CAP_LDATA	Λήψη των κατώτερων τιμών (8..0 ) ενός pixel.
DONE	Ενεργοποίηση του σήματος frame_done. Σηματοδοτεί το τέλος λήψης ενός frame – εικόνας.

**Πίνακας 6.3:** Περιγραφή καταστάσεων κυκλώματος καταγραφής εικόνας

Η λειτουργικότητα κάθε κατάστασης της μηχανής του σχήματος 6β είναι παρόμοια. Δίνοντας σαν είσοδο στο HDL Designer την περιγραφή του παραπάνω κυκλώματος και επιλέγοντας μετατροπή σε state machine, η μηχανή που προκύπτει απεικονίζεται παρακάτω. Παρατηρούμε ότι είναι παρόμοια με εξαίρεση τα junction που χρησιμοποιούνται τα οποία όμως δε διαφέρουν ουσιαστικά από το and που χρησιμοποιούμε στις παραπάνω καταστάσεις. Επίσης το HDL Designer αναγνωρίζει τις διαδικασίες τις οποίες τοποθετεί στη στήλη Concurrent statements καθώς και τα σήματα και τις εισόδους και εξόδους του κυκλώματος που τοποθετεί σε αντίστοιχες στήλες.

## Αρχιτεκτονική Συστήματος



Σχήμα 6.7: Μηχανή πεπερασμένων καταστάσεων παραγόμενη από το HDL Designer

### 6.4 Μέτρηση των χρωματικών συνιστωσών

Το σύστημα διεπαφής του αισθητήρα είναι υπεύθυνο για την μέτρηση των τιμών των κόκκινων, μπλε και πράσινων συνιστωσών κάθε pixel της εικόνας και για την αποστολή των τιμών αυτών στο υπόλοιπο σύστημα που υλοποιεί το λογισμικό και είναι υπεύθυνο για την επεξεργασία και την εφαρμογή αλγορίθμων επεξεργασίας εικόνας αλλά και των υπολογισμών των στατιστικών της εικόνας. Ο υπολογισμός των στατιστικών δεν υλοποιήθηκε σε υλικό εξαιτίας της αυξημένης πολυπλοκότητας και της αυξημένες απαιτήσεις όσον αφορά την υπολογιστική ισχύ για την εκτέλεση των διαιρέσεων.

Όπως προαναφέρθηκε η έξοδος του αισθητήρα είναι σε μορφή RGB (565), δηλαδή τα 16 bit ενός pixel διακρίνονται σε 5-bit για το κόκκινο, 6-bit για το πράσινο



και 5-bit για το μπλε. Το κύκλωμα μέτρησης των τιμών των χρωματικών συνιστωσών της εικόνας δέχεται σαν είσοδο ένα pixel μεγέθους 16 bit και έχει σαν έξοδο τις συνολικές τιμές των χρωματικών συνιστωσών της εικόνας.

## 6.5 Σύστημα μνήμης

Τα pixel που διαβάζονται από το κύκλωμα καταγραφής εικόνας αποθηκεύονται από το σύστημα μνήμης ώστε να είναι διαθέσιμα για ανάγνωση από το υπόλοιπο σύστημα. Το σύστημα αυτό αποτελείται από ένα κύκλωμα ελέγχου το οποίο ελέγχει την εγγραφή και ανάγνωση των pixel και ενεργοποιεί σήματα ελέγχου τα οποία σηματοδοτούν στο υπόλοιπο σύστημα την πληρότητα ή την κενότητα της μνήμης. Επιπλέον ενεργοποιούνται σήματα τα οποία ειδοποιούν το υπόλοιπο σύστημα ότι η μνήμη πρόκειται να φθάσει στο τέλος ανάγνωσης ή εγγραφής, για το καλύτερο συγχρονισμό της διεπαφής με το υπόλοιπο σύστημα.

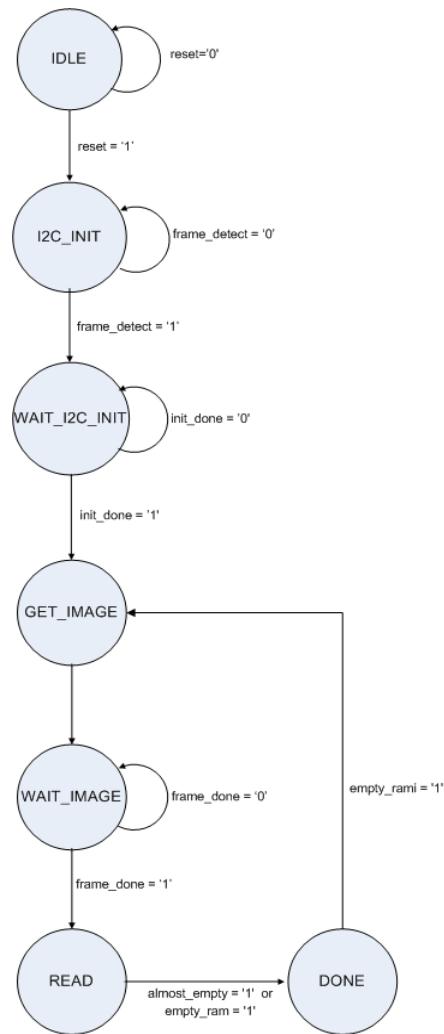
Τα στοιχεία της εικόνας αποθηκεύονται σε μια μνήμη τύπου ram από όπου είναι δυνατή η ταυτόχρονη ανάγνωση και εγγραφή για αύξηση της ταχύτητας και της απόδοσης του συστήματος.

## 6.6 Κύκλωμα ελέγχου συστήματος

Το πιο υψηλό ιεραρχικά κύκλωμα το οποίο ελέγχει το σύστημα είναι το κύκλωμα ελέγχου συστήματος. Είναι υπεύθυνο για τον συντονισμό και τη σωστή λήψη, την εγγραφή και ανάγνωση της εικόνας. Στέλνει σήμα ενεργοποίησης του I<sup>2</sup>C πρωτοκόλλου επικοινωνίας ώστε να γίνει η αρχικοποίηση των καταχωρητών του αισθητήρα και να είναι δυνατή η σύλληψη μιας εικόνας. Παίρνοντας το ρόλο του συστήματος καθορίζει την ανάγνωση της μνήμης όταν αυτή πρόκειται να γεμίσει.

Η μηχανή πεπερασμένων καταστάσεων (state machine) την οποία προτείνουμε για τον έλεγχο και συντονισμό της λειτουργίας του συστήματος καθώς και της διαδικασίας σύλληψης μιας εικόνας φαίνεται παρακάτω.

## Αρχιτεκτονική Συστήματος



**Σχήμα 6.8:** Μηχανή πεπερασμένων καταστάσεων κυκλώματος ελέγχου

Ο επόμενος πίνακας περιγράφει τη λειτουργικότητα της κάθε κατάστασης:

Όνομα κατάστασης	Λειτουργία
IDLE	Αναμονή έναρξης του συστήματος.
I2C_INIT	Ενεργοποίηση του σήματος ελέγχου start για την έναρξη αρχικοποίησης των καταχωρητών του αισθητήρα μέσω του I2C πρωτοκόλλου.
WAIT_I2C_INIT	Αναμονή ολοκλήρωσης της διαδικασίας αρχικοποίησης του αισθητήρα.
GET_IMAGE	Ενεργοποίηση του σήματος get_frame_data το οποίο σηματοδοτεί την έναρξη σύλληψης και λήψης ενός πλαισίου.

Όνομα κατάστασης	Λειτουργία
WAIT_IMAGE	Αναμονή ολοκλήρωσης διαδικασίας λήψης της εικόνας. Ενεργοποίηση του σήματος interrupt το οποίο θέτει τον αισθητήρα σε κατάσταση αναμονής εφόσον δεν έγινε σωστή λήψη της εικόνας .
READ	Αναμονή ολοκλήρωσης διαδικασίας ανάγνωσης των pixel.
DONE	Ενεργοποίηση του σήματος sm_rsti το οποίο αρχικοποιεί το κύκλωμα σύλληψης εικόνας υποδεικνύοντας την έναρξη λήψης νέας εικόνας από τον αισθητήρα.

**Πίνακας 6.4:** Περιγραφή καταστάσεων κυκλώματος ελέγχου

Επιπλέον ενεργοποιείται το σήμα ανάγνωσης rdin της εικόνας από το σύστημα εφόσον η μνήμη είναι γεμάτη ή έχει ολοκληρωθεί η λήψη ενός πλαισίου και συγχρόνως δεν έχει αδειάσει.

## 6.7 Έλεγχος καταχωρητών

Προτείνουμε τέλος την αρχιτεκτονική για τον έλεγχο και αρχικοποίηση των καταχωρητών του αισθητήρα, η οποία θα ελέγχεται από το λογισμικό που ακολουθεί τη διεπαφή του αισθητήρα. Αρχικά θα γίνεται έλεγχος εγγραφής ή ανάγνωσης κάποιου καταχωρητή και στη συνέχεια θα μεταβαίνουμε σε μια φάση ελέγχου εγγραφής όπου θα ελέγχεται εάν θα ακολουθήσει εγγραφή κάποιου καταχωρητή του πυρήνα του αισθητήρα (Core) οπότε και μεταβαίνουμε σε αντίστοιχη κατάσταση ή κάποιου καταχωρητή του επεξεργαστή εικόνας (Image Flow Processor) του αισθητήρα οπότε και μεταβαίνουμε στην αντίστοιχη κατάσταση. Εάν αιτηθεί ανάγνωση μεταβαίνουμε αντίστοιχα σε καταστάσεις ανάγνωσης καταχωρητή του πυρήνα ή του επεξεργαστή εικόνας του αισθητήρα. Σε κάθε περίπτωση θα υπάρχει κατάσταση αναμονής για την ολοκλήρωση της ανάγνωσης ή της εγγραφής του/ των καταχωρητών.



## Περιγραφή γλώσσας υλικού

### Λειτουργική προσομοίωση

Το πρώτο βήμα της ροής πληροφορίας όπως απεικονίζεται στα σχήματα 5.1 και 5.3 είναι η περιγραφή του κυκλώματος σε γλώσσα υλικού (VHDL) και η λειτουργική προσομοίωση, όπου γίνεται συνεχής έλεγχος της σχεδίασης μέχρι να παρατηρήσουμε την επιθυμητή. Στο στάδιο αυτό ο σχεδιαστής δαπανά περίπου το 50% του συνολικού χρόνου ολοκλήρωσης της σχεδίασης. Πλεονέκτημα της περιγραφής σε γλώσσα υλικού παρά σε σχηματικά διαγράμματα του κυκλώματος είναι η δυνατότητα απευθείας προσομοίωσης πριν τη σύνθεση του κυκλώματος. Στο παρόν λοιπόν κεφάλαιο αναφέρονται τα εργαλεία που χρησιμοποιήθηκαν στο στάδιο αυτό και παρουσιάζονται αναλυτικά τα αποτελέσματα.

## 7.1 Εργαλεία

Στην υλοποίηση της αρχιτεκτονικής του συστήματος χρησιμοποιήθηκαν τα εξής εργαλεία:

- **Matlab 7.0**

Σε αυτό έγινε η υλοποίηση και η προσομοίωση των αλγορίθμων επεξεργασίας της εικόνας, εξισορρόπησης λευκού και αυτόματης έκθεσης. Επιπλέον χρησιμοποιήθηκε για την ανάγνωση μιας εικόνας και της εγγραφής των στοιχείων της σε αρχείο το οποίο έπειτα χρησιμοποιήθηκε στην προσομοίωση του κυκλώματος [3].

- **ModelSim SE 6.0a**

Σε αυτό το εργαλείο έγινε η υλοποίηση και η περιγραφή της αρχιτεκτονικής του συστήματος σε γλώσσα υλικού (vhdl). Επίσης χρησιμοποιήθηκε για την προσομοίωση και τον έλεγχο της σωστής λειτουργίας του συνολικού κυκλώματος και των επιμέρους κυκλωμάτων [38].

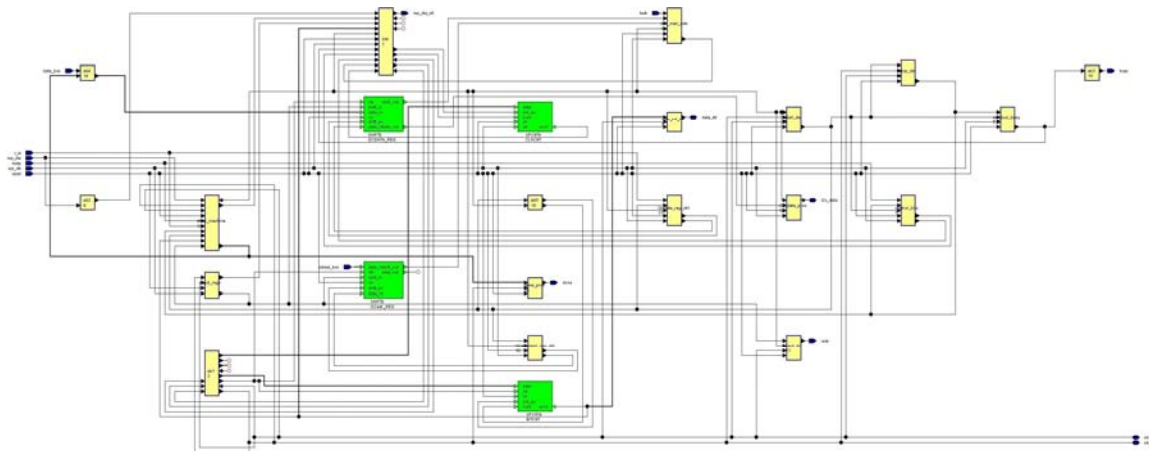
- **HDL Designer 2004.1b**

Πρόκειται για ένα νέο εργαλείο το οποίο δίνει την δυνατότητα για μετατροπή ενός κυκλώματος το οποίο έχει περιγραφεί σε γλώσσα υλικού, σε σχηματική μορφή (block διάγραμμα) και το αντίστροφο [39]. Επιπλέον δίνει την δυνατότητα καλύτερης διαχείρισης και οργάνωσης ενός σχεδίου - project και προσφέρει την δυνατότητα διασύνδεσης με τα εργαλεία ModelSim SE 6.0a και Leonardo Spectrum για ολοκληρωμένη υλοποίηση ενός κυκλώματος. Χρησιμοποιήθηκε για την δημιουργία των σχηματικών διαγραμμάτων και ελέγχθηκε ο κώδικας που παράγει, δίνοντας σαν είσοδο μηχανές πεπερασμένων καταστάσεων, με πολύ καλά αποτελέσματα. Επιπλέον δίνει τη δυνατότητα για χρήση κάποιων έτοιμων βασικών υλοποιήσεων όπως μνήμες, fifo και ram.

## 7.2 Πρωτόκολλο επικοινωνίας I<sup>2</sup>C

Το συνολικό κύκλωμα που υλοποιεί το πρωτόκολλο επικοινωνίας απεικονίζεται στο επόμενο σχήμα. Εκτός των άλλων υλοποιήθηκε ένας παράλληλος ολισθητής ο οποίος καλείται δυο φορές από το κύκλωμα, για την ολίσθηση της διεύθυνσης του αισθητήρα καθώς και της διεύθυνσης του κάθε καταχωρητή. Επιπλέον χρησιμοποιούνται μετρητές για την μέτρηση των bit σε κάθε μεταφορά και τη μέτρηση των παλμών του ρολογιού κατά τη δημιουργία των περιόδων του scl από τον master. Τα υπόλοιπα στοιχεία του παρακάτω κυκλώματος αναπαριστούν μηχανές πεπερασμένης κατάστασης που αναφέρθηκαν στην αρχιτεκτονική και διαδικασίες οι οποίες αποδίδουν τιμές στις εξόδους του κυκλώματος. Κάποιες από τις διαδικασίες αυτές επιτελούν σημαντικές λειτουργίες. Ανιχνεύουν σήματα έναρξης ή τερματισμού του πρωτοκόλλου, θέτουν τις σωστές τιμές στη γραμμή sda και αναγνωρίζουν τη λήψη επιβεβαίωσης από τον αισθητήρα.

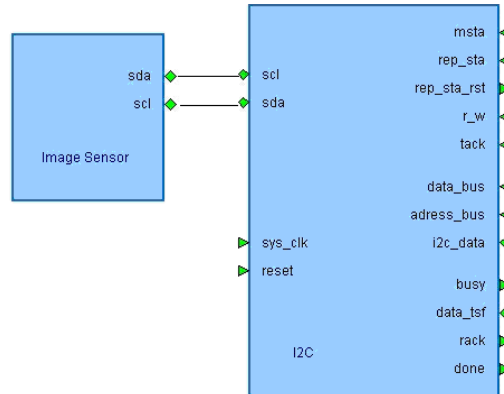
Η ανίχνευση έναρξης γίνεται ως εξής. Η διαδικασία που είναι υπεύθυνη παρακολουθεί τις γραμμές scl και sda και όταν αντιληφθεί scl='1' και sda='0' τότε παράγει σήμα ανίχνευσης έναρξης, σε αντίθετη περίπτωση παράγεται σήμα τερματισμού της λειτουργίας (scl='0' και sda='0'). Όσον αφορά επίσης το bit r\_w, αυτό τίθεται ίσο με τη μονάδα εάν πρόκειται να γίνει εγγραφή οπότε και το τελευταίο ψηφίο της διεύθυνσης της slave συσκευής θα είναι 0, ενώ τίθεται ίσο με το μηδέν εάν πρόκειται να γίνει ανάγνωση οπότε και το τελευταίο ψηφίο της διεύθυνσης της slave συσκευής θα είναι 1.



Σχήμα 7.1: Συνολικό κύκλωμα πρωτοκόλλου επικοινωνίας I<sup>2</sup>C

## Λειτουργική προσομοίωση

Το επόμενο σχήμα απεικονίζει το block διάγραμμα του κυκλώματος που υλοποιεί το πρωτόκολλο επικοινωνίας με τις εξόδους και εισόδους προς το σύστημα.



**Σχήμα 7.2:** Block διάγραμμα κυκλώματος πρωτοκόλλου επικοινωνίας I<sup>2</sup>C

Στον επόμενο πίνακα περιγράφονται αναλυτικά οι πύλες του κυκλώματος.

Πύλη	Τύπος	Μέγεθος	Περιγραφή
clk	Είσοδος	1 - bit	Ρολόι συστήματος
reset	Είσοδος	1 - bit	Αρχικοποίηση του συστήματος
sda	Είσοδος Έξοδος	1 - bit	Μεταφορά δεδομένων από και προς τον αισθητήρα
scl	Είσοδος Έξοδος	1 - bit	Ρολόι που παράγεται από τον master και ελέγχει την μετάδοση.
msta	Είσοδος	1 - bit	Όταν ενεργοποιείται από τον master, τίθεται ίσο με τη μονάδα τότε σηματοδοτεί την έναρξη της μετάδοσης ενώ όταν τίθεται στο μηδέν δηλώνει το τέλος αυτής
tack	Είσοδος	1 - bit	Σήμα επιβεβαίωσης. Ενεργοποιείται από τον master όταν είναι σε κατάσταση receiver για να σηματοδοτήσει στον αισθητήρα ότι λήφθηκαν τα δεδομένα.
rep_sta	Είσοδος	1 - bit	Δηλώνει επαναλαμβανόμενη έναρξη όταν μια μεταφορά τελειώσει
rep_sta_rst	Έξοδος	1 - bit	Δηλώνει ότι το σήμα επαναλαμβανόμενης έναρξης πρέπει να μηδενιστεί γιατί δεν λήφθηκε σε κατάλληλη στιγμή.
r_w	Είσοδος	1 - bit	Δηλώνει το mode στο οποίο θα λειτουργήσει το σύστημα. Αν είναι ίσο με μηδέν τότε ο master θα λάβει δεδομένα κάποιου καταχωρητή ενώ αν είναι ίσο με τη μονάδα θα στείλει δεδομένα σε κάποιο καταχωρητή. Ουσιαστικά είναι το 8 <sup>ο</sup> bit της διεύθυνσης της slave συσκευής.
data_bus	Είσοδος	16 - bits	Μεταφορά δεδομένων από το σύστημα στην διεπαφή του πρωτοκόλλου. Μεταφέρει τα δεδομένα που θα προγραμματίσουν τον εκάστοτε καταχωρητή

## Λειτουργική προσομοίωση

Πύλη	Τύπος	Μέγεθος	Περιγραφή
adress_bus	Είσοδος	16 - bits	Μεταφορά της διεύθυνσης του αισθητήρα και της διεύθυνσης του καταχωρητή
i2c_data	Έξοδος (inout)	24 - bits	Μεταφορά των δεδομένων ενός καταχωρητή προς το σύστημα.
busy	Έξοδος	1 - bit	Δηλώνει ότι η αρτηρία μεταφοράς του πρωτοκόλλου είναι απασχολημένη
data_tsf	Έξοδος	1 - bit	Δηλώνει το τέλος μιας μεταφοράς 8-bit
rack	Έξοδος	1 - bit	Δηλώνει την λήψη σήματος επιβεβαίωσης από τον αισθητήρα
done	Έξοδος	1 - bit	Δηλώνει το τέλος μεταφοράς των δεδομένων προς τον καταχωρητή, τέλος προγραμματισμού καταχωρητή.

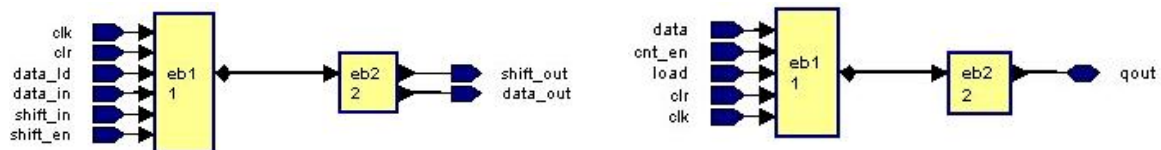
**Πίνακας 7.1:** Περιγραφή πυλών κυκλώματος πρωτοκόλλου επικοινωνίας I<sup>2</sup>C

Για την παραγωγή του σήματος done χρησιμοποιείται μετρητής ο οποίος μετράει τις μεταφορές που έχουν γίνει στην γραμμή sda και όταν συμπληρωθεί ο αριθμός που χρειάζεται για να προγραμματιστεί ο καταχωρητής ενεργοποιείται το σήμα done.

Χρησιμοποιούνται δύο διαφορετικές αρτηρίες data\_bus και adress\_bus για τη μεταφορά των δεδομένων και της διεύθυνσης αντίστοιχα ώστε να στέλνονται ταυτόχρονα από το σύστημα (λογισμικό) στη διεπαφή του πρωτοκόλλου η διεύθυνση και τα δομένα προγραμματισμού του εκάστοτε καταχωρητή.

Το υπόλοιπο σύστημα θα στέλνει τα σήματα έναρξης ή τερματισμού όταν είναι σε transmit mode και όταν λαμβάνει το σήμα done θα μπορεί να ενεργοποιήσει το σήμα επαναλαμβανόμενης έναρξης για να προγραμματιστεί επόμενος καταχωρητής.

Στο επόμενο σχήμα απεικονίζονται τα κυκλώματα του ολισθητή και του μετρητή που χρησιμοποιήθηκαν στην υλοποίηση.



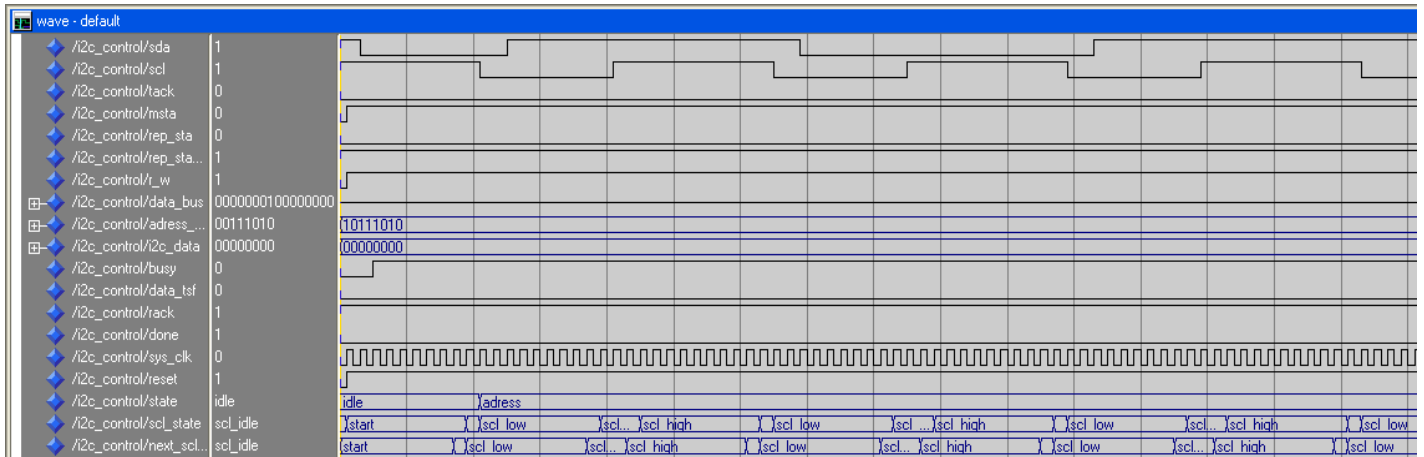
**Σχήμα 7.3:** (α) Κύκλωμα ολισθητή, (β) Κύκλωμα μετρητή

Το κύκλωμα υλοποιήθηκε σε vhd1 και ελέγχθηκε για τη σωστή λειτουργία με τη χρήση του Modelsim. Παρακάτω φαίνονται τα αποτελέσματα της προσομοίωσης. Αρχικά στο σχήμα 4 προσομοιώνεται η αποστολή δεδομένων. Στο σχήμα 4α απεικονίζεται στιγμιότυπο της αρχής λειτουργίας του κυκλώματος όπου στέλνεται η διεύθυνση εγγραφής του αισθητήρα, 0XBA – 10111010. Στη συνέχεια στο σχήμα 4β

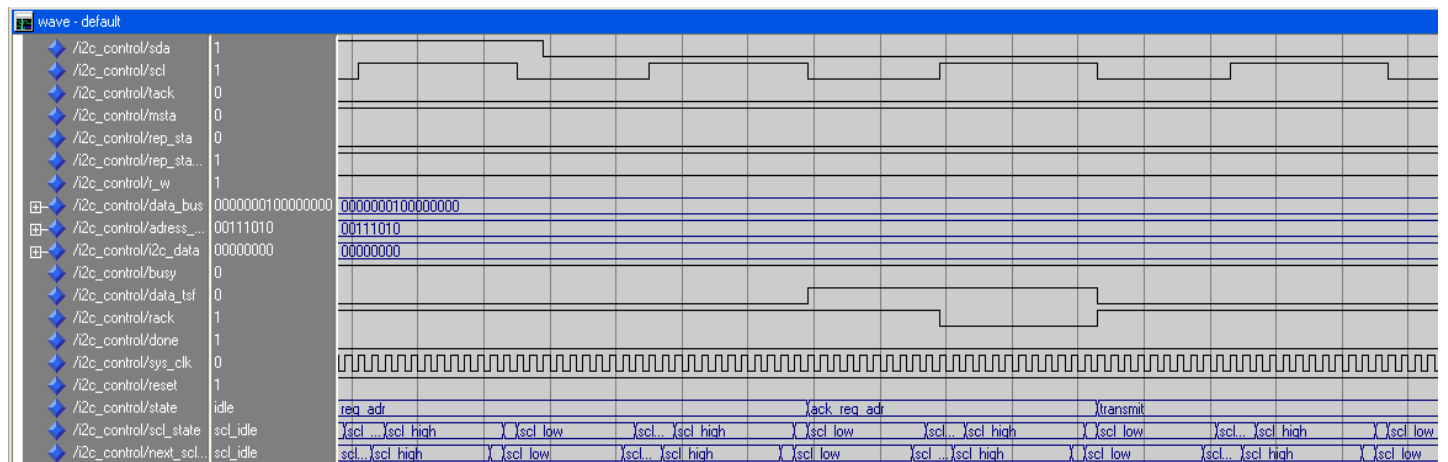


## Λειτουργική προσομοίωση

στέλλονται τα δεδομένα που προγραμματίζουν τον καταχωρητή του αισθητήρα να παράγει την έξοδο σε μορφή 565 RGB (0000000100000000). Τέλος στο σχήμα 4γ φαίνεται η αναμονή λήψης επιβεβαίωσης και η επανέναρξη της επικοινωνίας λαμβάνοντας το σήμα `rep_start`. Παρατηρούμε ότι το `r_w` είναι 1 δηλώνοντας ότι θα ακολουθήσει εγγραφή.

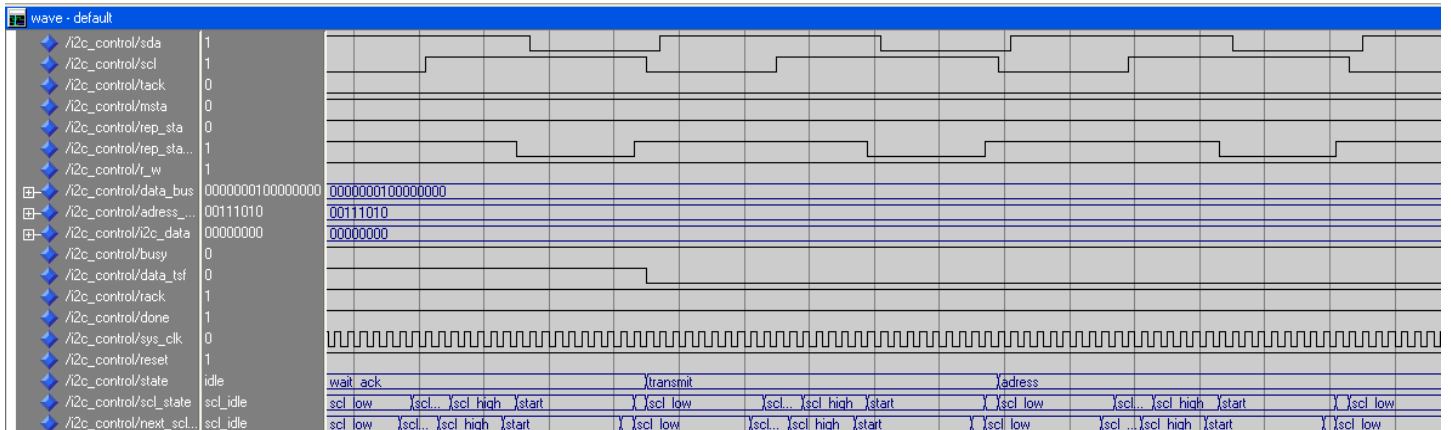


(α): Έναρξη λειτουργίας, αποστολή διεύθυνσης



(β): Λήψη επιβεβαίωσης, αποστολή δεδομένων

## Λειτουργική προσομοίωση



(γ): Επαναλαμβανόμενη έναρξη

**Σχήμα 7.4:** Αποτελέσματα προσομοίωσης αποστολής δεδομένων

Στα επόμενα σχήματα απεικονίζονται στιγμιότυπα του παραθύρου list όπου φαίνονται αναλυτικά οι τιμές σε κάθε χρονική στιγμή. Στο σχήμα 5α απεικονίζεται η λήψη επιβεβαίωσης και η έναρξη αποστολής της διεύθυνσης του καταχωρητή. Στο συγκεκριμένο παράδειγμα προγραμματίζεται ο καταχωρητής R58:1 – 0X13A (00111010), υπεύθυνος για τον τύπο εξόδου του αισθητήρα, για να παράγει την έξοδο σε μορφή RGB 565. Στο επόμενο σχήμα απεικονίζεται η αποστολή των 8 – bit δεδομένων. Σημειώνουμε ότι στην υλοποίηση του κυκλώματος επιλέξαμε να έχουμε είσοδο από το σύστημα 15-bit για τη μεταφορά των δεδομένων του καταχωρητή τα οποία με βάση ένα μετρητή και γνωρίζοντας τον αριθμό των μεταφορών (ο καταχωρητής απαιτεί 2 μεταφορές για τον προγραμματισμό του) στέλνουμε πρώτα τα πιο σημαντικά bit και στη συνέχεια τα υπόλοιπα. Τέλος στο επόμενο σχήμα φαίνεται η λήξη της επικοινωνίας με την ανίχνευση του σήματος stop.

ns	/i2c_control/sda	/i2c_control/rep_sta_rst	/i2c_control/address_bus	/i2c_control/done	/i2c_control/state	/i2c_control/scl
4170 +1	0 1 0 1 0	1 1	0000000100000000	00111010	00000000	1 1 0 0
4170 +2	0 0 0 1 0	1 1	0000000100000000	00111010	00000000	1 1 0 0
4170 +3	0 0 0 1 0	1 1	0000000100000000	00111010	00000000	1 0 1 0
4170 +4	0 0 0 1 0	1 1	0000000100000000	00111010	00000000	1 0 1 0
4170 +5	0 0 0 1 0	1 1	0000000100000000	00111010	00000000	1 0 1 0
4180 +0	0 0 0 1 0	1 1	0000000100000000	00111010	00000000	1 0 1 0

(α): Λήψη επιβεβαίωσης, αποστολή διεύθυνσης καταχωρητή

## Λειτουργική προσομοίωση

ns	delta	/i2c_control/state	/i2c_control/scl_in	/i2c_control/sda_out	/i2c_control/shift_reg	/i2c_control/shift_addr	/i2c_control/shift_out	/i2c_control/shift_addr_reg	/i2c_control/shift_data_reg	/i2c_control/shift_data	
8130	+0	1 1	ack_reg_addr	scl_low_edge	scl_low	1 0 1 0	0 0 0 0 1	00000001	ZZZZZZZZ	Z 0 1	00111010
8130	+1	1 1	ack_reg_addr	scl_low	scl_low	1 0 0 0	0 0 0 0 1	00000001	ZZZZZZZZ	Z 0 1	00111010
8130	+2	1 1	ack_reg_addr	scl_low	scl_low	1 0 0 0	0 0 0 0 1	00000001	ZZZZZZZZ	Z 0 1	00111010
8130	+3	1 1	transmit	scl_low	scl_low	1 0 0 1	0 0 0 0 1	00000001	ZZZZZZZZ	Z 0 1	00111010
8130	+4	1 1	transmit	scl_low	scl_low	1 0 0 1	0 0 0 0 1	00000001	ZZZZZZZZ	Z 0 1	00111010
8130	+5	1 1	transmit	scl_low	scl_low	1 0 0 1	0 0 0 0 1	00000001	00000001	0 0 1	00111010
8140	+0	0 1	transmit	scl_low	scl_low	1 0 0 1	0 0 0 0 1	00000001	00000001	0 0 1	00111010
8150	+0	1 1	transmit	scl_low	scl_low	1 0 0 1	0 0 0 0 1	00000001	00000001	0 0 1	00111010
8150	+1	1 1	transmit	scl_low	scl_low	0 0 0 1	0 0 0 0 1	00000001	00000001	0 1 0	00111010
8150	+2	1 1	transmit	scl_low	scl_low	0 0 0 1	0 0 0 0 1	00000001	00000001	0 1 0	00111010
8160	+0	0 1	transmit	scl_low	scl_low	0 0 0 1	0 0 0 0 1	00000001	00000001	0 1 0	00111010

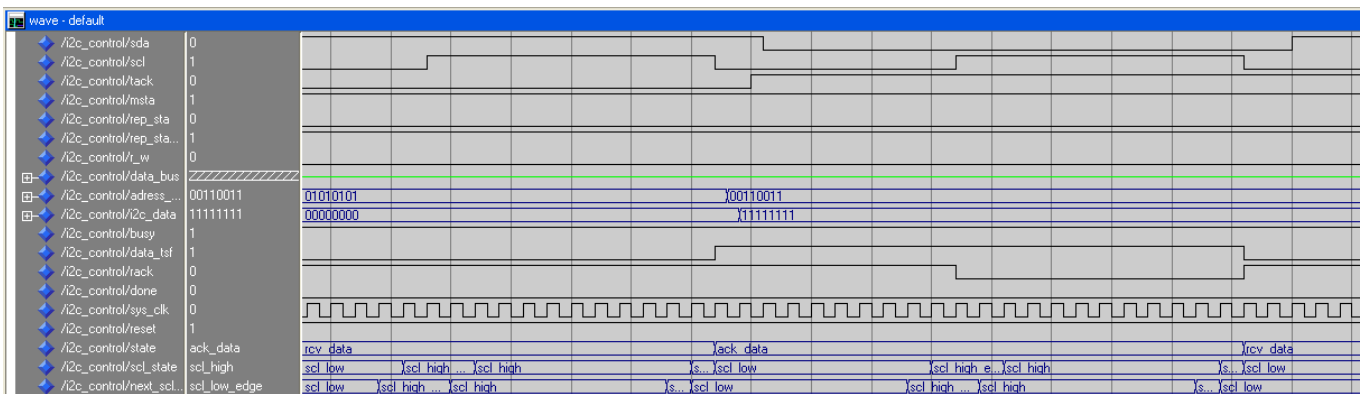
### (β): Αποστολή 8-bit δεδομένων

ns	delta	state	/i2c_control/scl_in	/i2c_control/sda_out	/i2c_control/shift_reg	/i2c_control/shift_addr	/i2c_control/shift_out	/i2c_control/shift_addr_reg	/i2c_control/shift_data_reg	/i2c_control/shift_data	
28790	+2	idle	stop	wait	stop	wait	1 1 1 0	1 1	1 1 0	00000001 10000000 1 0 0	00111010 0 0 1 0
28800	+0	idle	stop	wait	stop	wait	1 1 1 0	1 1	1 1 0	00000001 10000000 1 0 0	00111010 0 0 1 0
28810	+0	idle	stop	wait	stop	wait	1 1 1 0	1 1	1 1 0	00000001 10000000 1 0 0	00111010 0 0 1 0
28810	+2	idle	stop	wait	stop	wait	1 1 1 0	1 1	1 1 0	00000001 10000000 1 0 0	00111010 0 0 1 0

### (γ): Ολοκλήρωση επικοινωνίας

**Σχήμα 7.5:** Αποτελέσματα προσομοίωσης αποστολής δεδομένων παραθύρου list

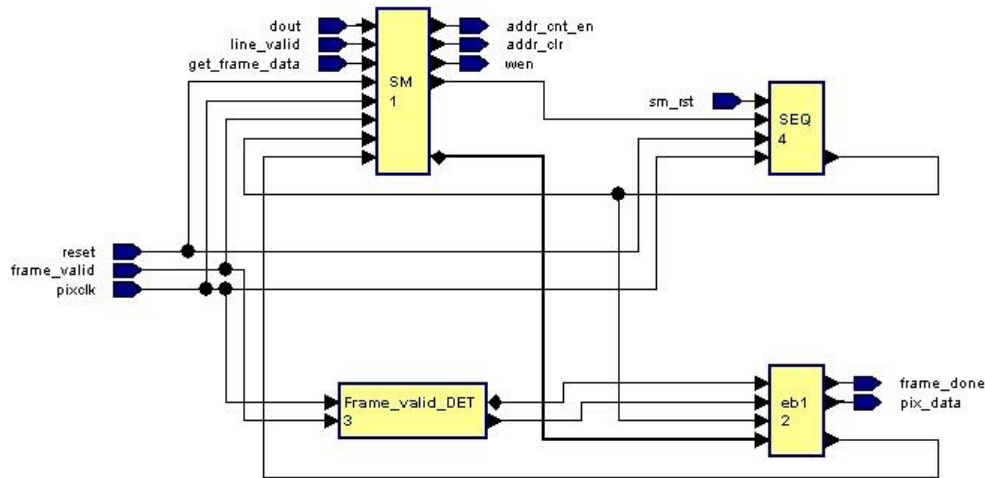
Στο επόμενο σχήμα προσομοιώνεται η λήψη δεδομένων από κάποιον καταχωρητή του αισθητήρα. Η διαδικασία αποστολής της διεύθυνσης του αισθητήρα και της διεύθυνσης του καταχωρητή που επιθυμούμε να προγραμματίσουμε είναι παρόμοια με αυτή του σχήματος 4α και 4β. Η μόνη διαφορά είναι ότι η διεύθυνσης ανάγνωσης του αισθητήρα είναι 0XBB – 10111011. Στο σχήμα 6 φαίνεται η αποστολή επιβεβαίωσης για τα δεδομένα (11111111) που παραλήφθηκαν από τον καταχωρητή.



**Σχήμα 7.6:** Λήψη δεδομένων



## Λειτουργική προσομοίωση



Σχήμα 7.7: Block διάγραμμα κυκλώματος καταγραφής εικόνας

Στον επόμενο πίνακα περιγράφονται αναλυτικά οι πύλες του κυκλώματος.

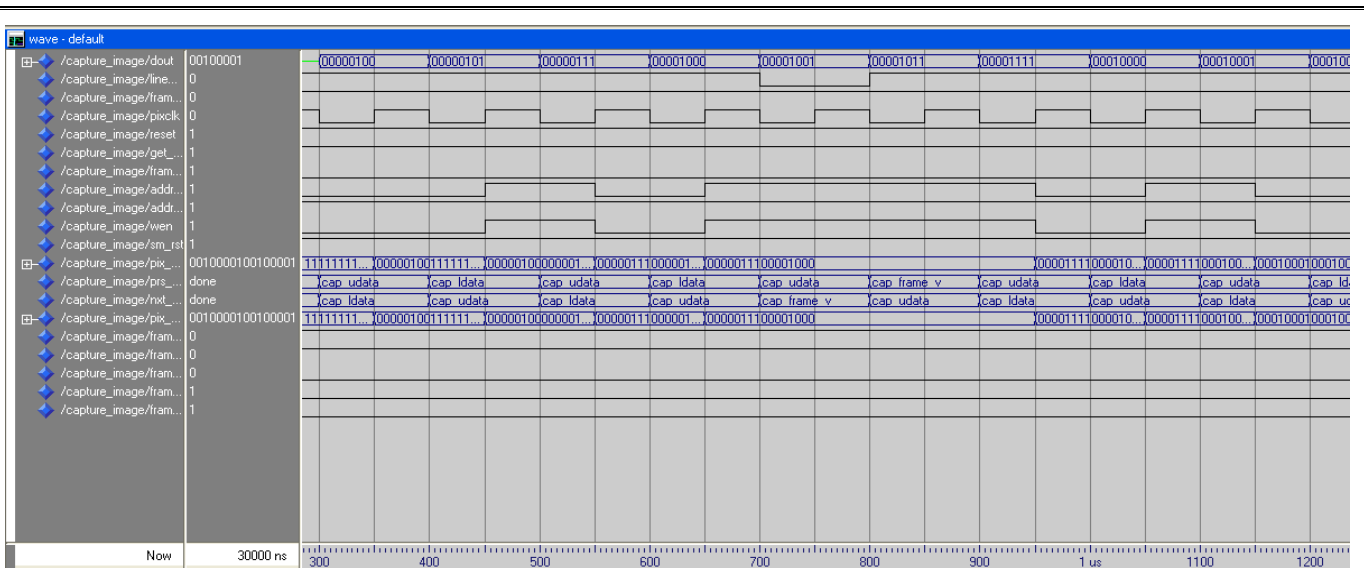
Πύλη	Τύπος	Μέγεθος	Περιγραφή
dout	Είσοδος	8 - bit	Μεταφορά των 8-bit κάθε pixel
line_valid	Είσοδος	1 - bit	Έγκυρη γραμμή δεδομένων (HSYNC)
frame_valid	Είσοδος	1 - bit	Έγκυρο πλαίσιο δεδομένων (VSYNC)
pixclk	Είσοδος	1 - bit	Ρολόι του αισθητήρα που συγχρονίζει την μεταφορά
Reset	Είσοδος	1 - bit	Αρχικοποίηση του συστήματος
get_frame_data	Είσοδος	1 - bit	Ενεργοποίηση από το σύστημα, σηματοδοτεί την έναρξη λήψης μιας εικόνας
frame_done	Έξοδος	1 - bit	Τέλος λήψης πλαισίου
sm_rst	Είσοδος	1 - bit	Ενεργοποίηση από το σύστημα, επανέναρξη λήψης ενός πλαισίου
addr_cnt_en	Έξοδος	1 - bit	Ενεργοποίηση αύξησης διεύθυνσης της μνήμης
addr_clr	Έξοδος	1 - bit	Μηδενισμός των διευθύνσεων της μνήμης πριν την αποθήκευση νέου πλαισίου εφόσον ολοκληρώθηκε η ανάγνωση του προηγούμενου
wen	Έξοδος	1 - bit	Ενεργοποίηση εγγραφής της μνήμης
pix_data	Έξοδος	16 - bit	Μεταφορά των pixel στην μνήμη (16-bit 565RGB)

Πίνακας 7.3: Περιγραφή πυλών κυκλώματος

Το κύκλωμα υλοποιήθηκε σε vhdl και ελέγχθηκε για τη σωστή λειτουργία με τη χρήση του Modelsim. Παρακάτω φαίνονται τα αποτελέσματα της προσομοίωσης. Στο σχήμα 8 φαίνεται η σωστή λειτουργία του κυκλώματος και οι μεταβάσεις στις

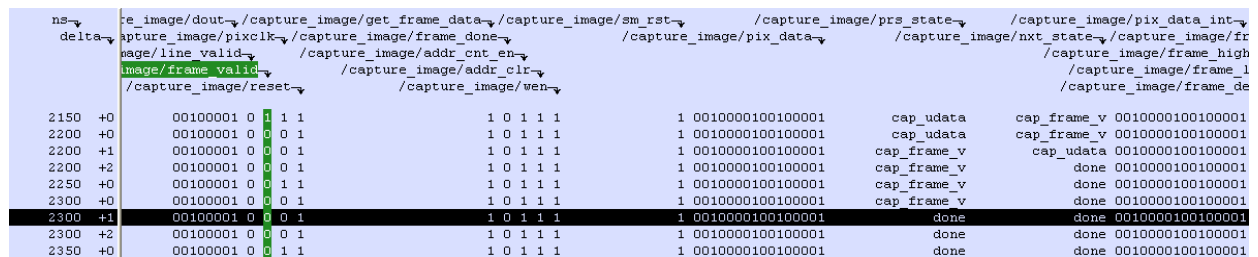
## Λειτουργική προσομοίωση

επιθυμητές καταστάσεις. Η έξοδος του αισθητήρα είναι 8-bit κάθε φορά και το συγκεκριμένο κύκλωμα δημιουργεί pixel 16-bit συνενώνοντας τα 8-bit που δέχεται κάθε φορά, πρώτα έρχονται τα πιο σημαντικά bit. Στη συνέχεια στέλνει κάθε pixel στη μνήμη ενεργοποιώντας τα σήματα `addr_cnt_en` και `wen` όπως φαίνεται για παράδειγμα στη χρονική στιγμή 450. Επιπλέον στη χρονική στιγμή 700 το `line_valid` γίνεται μηδέν, δηλαδή η γραμμή έχει τελειώσει και το κύκλωμα μεταβαίνει στην κατάσταση `CAP_FRAME_V` για να συλλέξει την επόμενη γραμμή του πλαισίου.



Σχήμα 7.8: Καταγραφή εικόνας

Στο σχήμα 9 τα σήματα `line_valid` και `frame_valid` έχουν τεθεί ίσα με το μηδέν σηματοδοτώντας το τέλος λήψης του πλαισίου και το κύκλωμα μεταβαίνει στην κατάσταση ολοκλήρωσης πλαισίου, `DONE`.



Σχήμα 7.9: Ολοκλήρωση λήψης πλαισίου

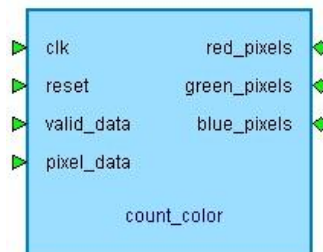
Το do αρχείο που εκτελούμε φαίνεται παρακάτω (παρατηρούμε ότι η περίοδος έχει οριστεί στα 74074 ps )

Βήμα	Εντολές
1 <sup>ο</sup>	vlib work vmap work work vcom capture_image.vhd vsim capture_image
2 <sup>ο</sup>	force dout ZZZZZZZZ 0,00000100 300,00000101 400,00000111 500,00001000 600,00001001 700,00001011 800,00001111 900,00010000 1000,00010001 1100,00010011 1200,00010111 1300,00011111 1400  force line_valid 0 0,1 50,1 200,1 250,1 300,1 400,0 700,1 800,1 900,0 2100,0 2200 force frame_valid 0 0,1 50,1 200,1 250,1 2100,0 2200 force pixclk 0 0,1 37037 -repeat 74074 force reset 0 0,1 50,1 100,1 200 force get_frame_data 0 0,1 50,1 200,1 250 force sm_rst 1 0,1 50,1 200,1 250
3 <sup>ο</sup>	add list * add wave * run 3000000

Πίνακας 7.4: : Παράδειγμα .do αρχείου για προσομοίωση καταγραφής εικόνας

## 7.4 Μέτρηση των χρωμάτων

Για την μέτρηση των τιμών των χρωματικών συνιστωσών της εικόνας υλοποιήθηκε ένα ανεξάρτητο κύκλωμα το οποίο δέχεται σαν είσοδο τα pixels της εικόνας σε κάθε άνοδο του ρολογιού και μετρά τις τιμές του μπλε, κόκκινου και πράσινου κάθε φορά με χρήση μετρητών. Κάθε pixel είναι σε μορφή RGB (565), δηλαδή τα 16 bit του pixel διακρίνονται σε 5-bit για το κόκκινο, 6- bit για το πράσινο και 5-bit για το μπλε. Στο επόμενο block διάγραμμα απεικονίζεται το κύκλωμα μέτρησης των χρωμάτων.



Σχήμα 7.10: Block διάγραμμα κυκλώματος μέτρησης χρωμάτων

Στον επόμενο πίνακα περιγράφονται αναλυτικά οι πύλες του κυκλώματος.

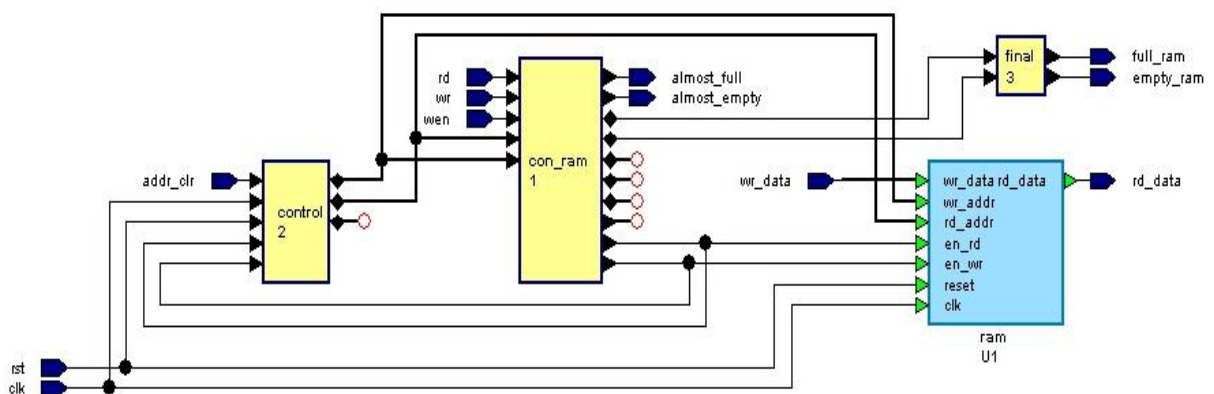
Πύλη	Τύπος	Μέγεθος	Περιγραφή
clk	Είσοδος	1 - bit	Ρολόι συστήματος (αισθητήρα)
reset	Είσοδος	1 - bit	Αρχειοποίηση του συστήματος
valid_data	Είσοδος	1 - bit	Έγκυρα δεδομένα στην αρτηρία pixel_data
pixel_data	Είσοδος	16 - bits	Μεταφορά των pixel
red_pixels	Έξοδος (inout)	24 - bits	Σύνολο της τιμής της κόκκινης χρωματικής συνιστώσας των pixel της εικόνας
green_pixels	Έξοδος (inout)	25 - bits	Σύνολο της τιμής της πράσινης χρωματικής συνιστώσας των pixel της εικόνας
blue_pixels	Έξοδος (inout)	24 - bits	Σύνολο της τιμής της μπλε χρωματικής συνιστώσας των pixel της εικόνας

**Πίνακας 7.5:** Περιγραφή πυλών κυκλώματος μέτρησης χρωμάτων

Το κύκλωμα υλοποιήθηκε σε vhdl και ελέγχθηκε για τη σωστή λειτουργία με τη χρήση του Modelsim. Τα αποτελέσματα της προσομοίωσης και της σωστής λειτουργίας φαίνονται στα σχήματα που ακολουθούν.

## 7.5 Σύστημα μνήμης

Το σύστημα μνήμης αποτελείται από 2 κυκλώματα, το κύκλωμα ελέγχου της μνήμης και μια μνήμη ram για την αποθήκευση των pixel. Το συνολικό κύκλωμα που υλοποιήθηκε φαίνεται στο επόμενο σχήμα. Το στοιχείο control υλοποιεί τη διαδικασία ελέγχου των διευθύνσεων της μνήμης, αυξάνοντας ή ελαττώνοντας τις διευθύνσεις ανάλογα αν λαμβάνει χώρα εγγραφή ή ανάγνωση της μνήμης αντίστοιχα.

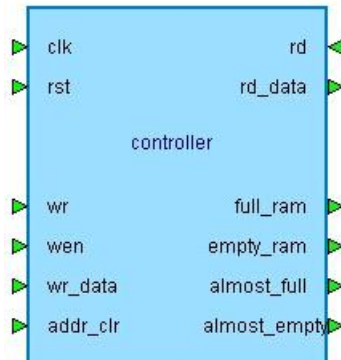


**Σχήμα 7.11:** Σύστημα μνήμης



### 7.5.1 Κύκλωμα ελέγχου

Στο επόμενο block διάγραμμα απεικονίζεται το κύκλωμα ελέγχου της μνήμης .



Σχήμα 7.12: Block διάγραμμα κυκλώματος ελέγχου της μνήμης

Στον επόμενο πίνακα περιγράφονται αναλυτικά οι πύλες του κυκλώματος.

Πύλη	Τύπος	Μέγεθος	Περιγραφή
clk	Είσοδος	1 - bit	Ρολόι συστήματος (αισθητήρα)
Rst	Είσοδος	1 - bit	Αρχικοποίηση του συστήματος
Rd	Είσοδος	1 - bit	Αίτηση για ανάγνωση της μνήμης από το σύστημα
addr_clr	Είσοδος	1 - bit	Μηδενισμός των διευθύνσεων μνήμης πριν αρχίσει η λήψη του πλαισίου. Ενεργοποιείται από το κύκλωμα σύλληψης εικόνας
wen	Είσοδος	1 - bit	Ενεργοποίηση εγγραφής, έγκυρα δεδομένα στην αρτηρία wr_data
wr	Είσοδος	1 - bit	Ενεργοποίηση αύξησης διεύθυνσης της μνήμης
wr_data	Είσοδος	16 - bits	Μεταφορά των pixel στην μνήμη
rd_data	Έξοδος	16 - bits	Μεταφορά των pixel στο υπόλοιπο σύστημα ύστερα από αίτηση ανάγνωσης
full_ram	Έξοδος	1 - bit	Δηλώνει γεμάτη μνήμη
empty_ram	Έξοδος	1 - bit	Δηλώνει άδεια μνήμη
almost_full	Έξοδος	1 - bit	Μνήμη σχεδόν γεμάτη
almost_empty	Έξοδος	1 - bit	Μνήμη σχεδόν άδεια

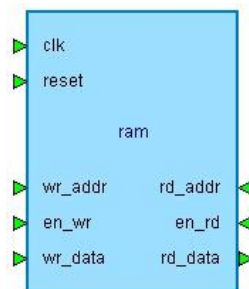
Πίνακας 7.6: Περιγραφή πυλών κυκλώματος ελέγχου της μνήμης

Στο κύκλωμα αυτό προσδιορίζεται δυναμικά με χρήση generic το μέγεθος της μνήμης και υλοποιούνται κάποιοι μετρητές. Ο wr\_i\_addr ο οποίος αυξάνεται στην άνοδο του ρολογιού, εάν είναι ενεργοποιημένο το σήμα wr\_i, το οποίο σηματοδοτεί την ενεργοποίηση εγγραφής της μνήμης, εάν τα σήματα εισόδου από το κύκλωμα σύλληψης εικόνας wen και wr είναι ενεργά και η μνήμη δεν είναι γεμάτη. Ο rd\_addr ο οποίος

αυξάνεται στην άνοδο του ρολογιού, εάν είναι ενεργοποιημένο το σήμα `rdi` το οποίο σηματοδοτεί την ενεργοποίηση ανάγνωσης της μνήμης, εάν το σήμα εισόδου από το σύστημα `rd` είναι ενεργό και η μνήμη δεν είναι άδεια. Επιπλέον παράγονται τα σήματα εξόδου `full_ram` και `empty_ram` τα οποία προσδιορίζουν εάν η μνήμη είναι γεμάτη ή άδεια αντίστοιχα.

### 7.5.2 Στοιχείο μνήμης

Στο στοιχείο της μνήμης γίνεται ουσιαστικά η αποθήκευση των pixel της κάθε εικόνας. Πρόκειται για μια διπλών πυλών μνήμη `ram` η οποία δίνει την δυνατότητα ταυτόχρονης εγγραφής και ανάγνωσης των pixel της εικόνας. Το μέγεθος της προσδιορίζεται δυναμικά από το κύκλωμα ελέγχου όπως προαναφέρθηκε. Το στοιχείο μνήμης αποθηκεύει σε κάθε θέση ένα pixel αυξάνοντας κάθε φορά την διεύθυνση αποθήκευσης κατά ένα. Η λειτουργία είναι παρόμοια με μια δομή FIFO, αφού ότι εγγράφεται πρώτα αυτό διαβάζεται πρώτο. Το block διάγραμμα του κυκλώματος φαίνεται παρακάτω.

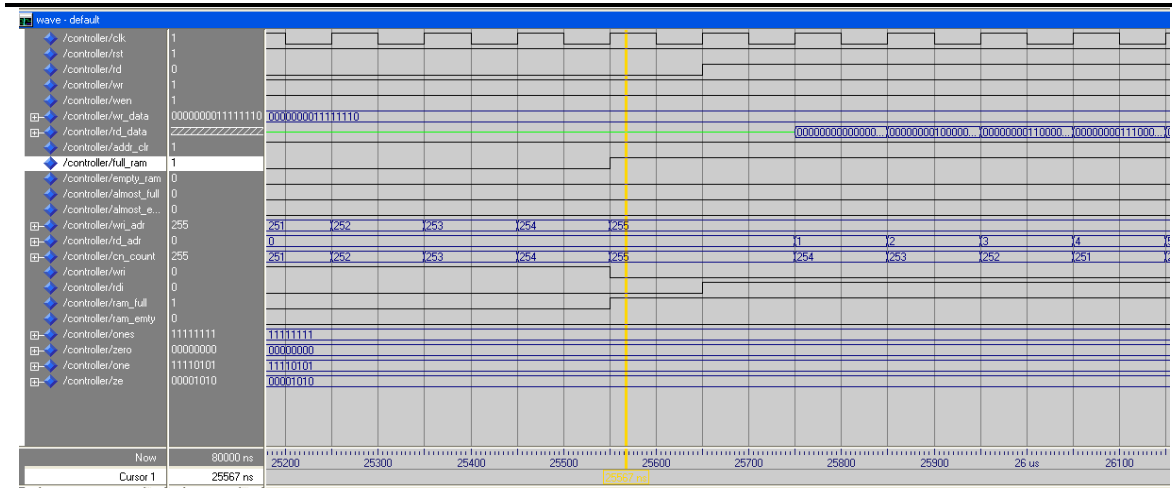


**Σχήμα 7.13:** Block διάγραμμα στοιχείου μνήμης

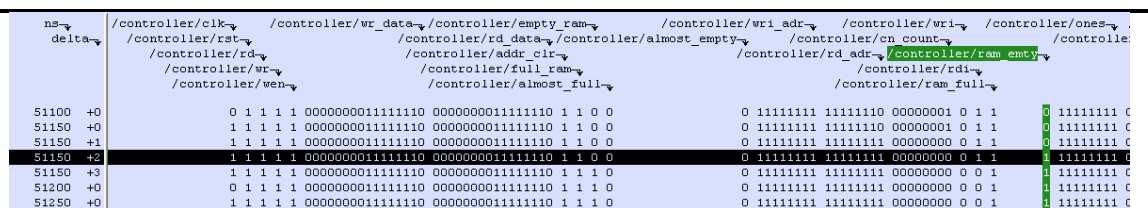
Αναφέρουμε ότι δοκιμάστηκε και η έτοιμη υλοποιημένη `ram` διπλών πυλών (`dual port`) του HDL Designer και η υλοποίηση είναι παρόμοια με αυτή του παραπάνω συστήματος. Το σύστημα μνήμης υλοποιήθηκε σε `vhdl` και ελέγχθηκε για τη σωστή λειτουργία με τη χρήση του `Modelsim`. Τα αποτελέσματα της προσομοίωσης και της σωστής λειτουργίας φαίνονται στα επόμενα σχήματα. Στο σχήμα 14 φαίνεται ότι η μνήμη, το μέγεθος της οποίας έχει οριστεί να είναι  $256 \times 16$ , να έχει γεμίσει και να έχει ενεργοποιηθεί το σήμα `ram_full`. Στο σχήμα 15 η μνήμη έχει αδειάσει αφού έχει γίνει

## Λειτουργική προσομοίωση

ανάγνωση όλων των στοιχείων της και έχει ενεργοποιηθεί το σήμα ram\_empty και μπορεί πλέον να επανεγγραφεί με το επόμενο πλαίσιο.

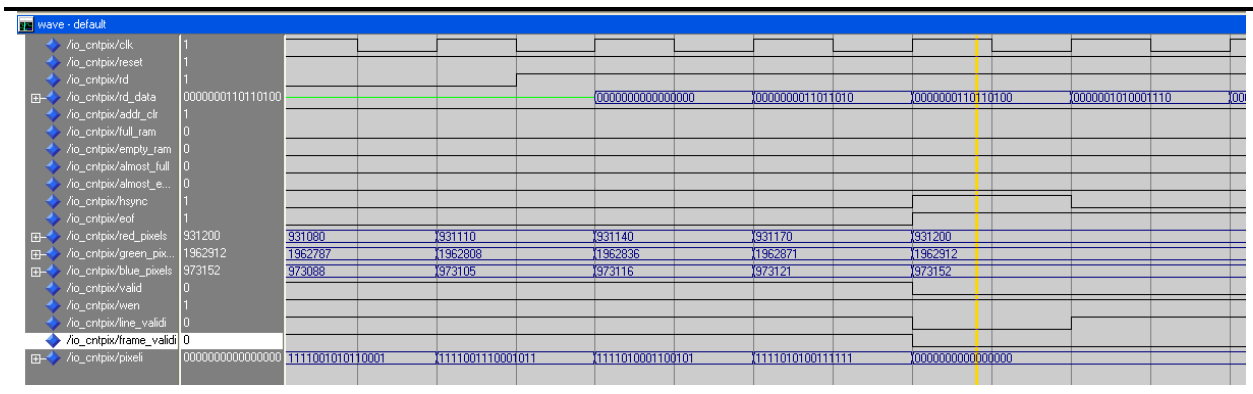


Σχήμα 7.14: Μνήμη γεμάτη



Σχήμα 7.15: Ολοκλήρωση ανάγνωσης μνήμης

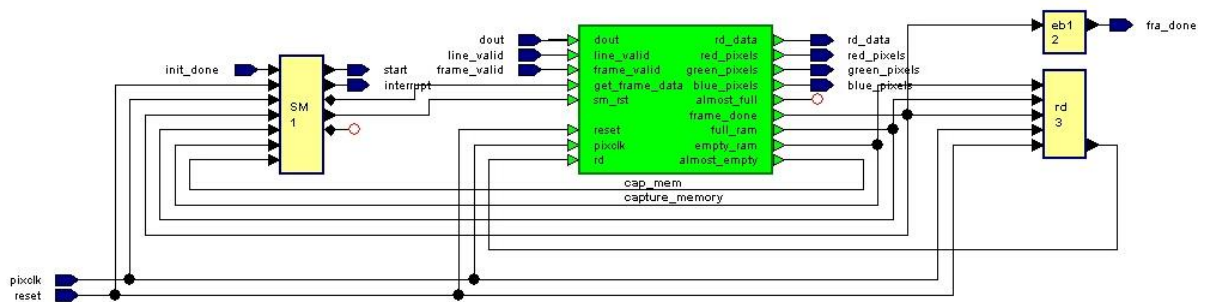
Στο σχήμα 16 τέλος γίνεται ανάγνωση μιας εικόνας, την οποία διαβάζουμε όπως περιγράφεται παρακάτω και εγγραφή της στη μνήμη (μεγέθους ίσου με αυτό της εικόνας) ενώ φαίνεται το τέλος γραμμής πλαισίου και το τέλος της εικόνας. Παρατηρούμε επίσης τους μετρητές των χρωματικών συνιστωσών και την έναρξη ανάγνωσης της εικόνας.



Σχήμα 7.16: Ολοκλήρωση εγγραφής πλαισίου και ανάγνωσης μνήμης

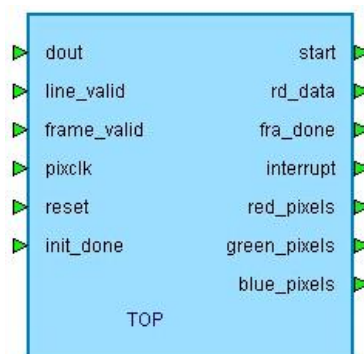
## 7.6 Κύκλωμα ελέγχου συστήματος

Το κύκλωμα αυτό είναι το πιο υψηλό ιεραρχικά και ενσωματώνει τα κυκλώματα σύλληψης εικόνας και το σύστημα μνήμης. Προσομοιώνει το υπόλοιπο σύστημα το οποίο θα συνδέεται στην διεπαφή και θα διαβάζει τα pixel από τη μνήμη. Έτσι το κύκλωμα αυτό παίρνοντας το ρόλο του συστήματος ελέγχει αν γέμισε η μνήμη ή αν τελείωσε η λήψη ενός πλαισίου και ανάλογα ενεργοποιεί το σήμα ανάγνωσης της μνήμης. Επιπλέον ελέγχει το I<sup>2</sup>C πρωτόκολλο και ενεργοποιεί τη λήψη ενός πλαισίου. Το συνολικό κύκλωμα απεικονίζεται στο επόμενο. Το SM 1 υλοποιεί τη μηχανή πεπερασμένων καταστάσεων του κυκλώματος ενώ το capture\_memory είναι το κύκλωμα που συνδέει το σύστημα μνήμης και το κύκλωμα σύλληψης εικόνας και απεικονίζεται στο σχήμα 21.



Σχήμα 7.17: Κύκλωμα ελέγχου συστήματος

Το block διάγραμμα του κυκλώματος με τις εισόδους και εξόδους φαίνεται παρακάτω.



Σχήμα 7.18: Block διάγραμμα κύκλωμα ελέγχου συστήματος

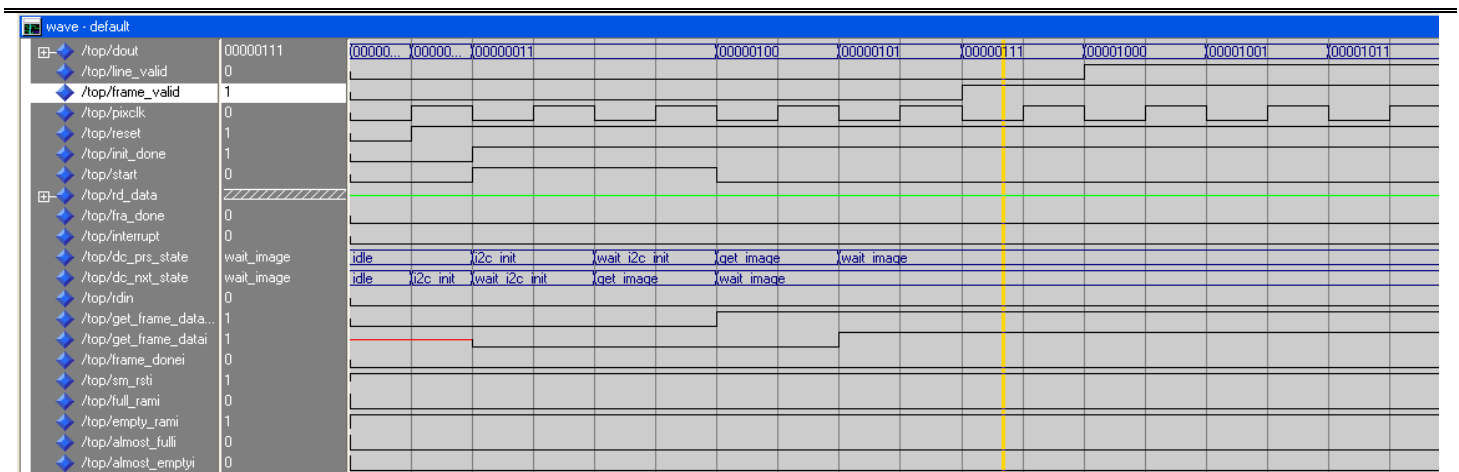
## Λειτουργική προσομοίωση

Στον επόμενο πίνακα περιγράφονται αναλυτικά οι πύλες του κυκλώματος.

Πύλη	Τύπος	Μέγεθος	Περιγραφή
pixclk	Είσοδος	1 - bit	Ρολόι συστήματος (αισθητήρα)
reset	Είσοδος	1 - bit	Αρχειοποίηση του συστήματος
dout	Είσοδος	8 - bits	Μεταφορά εξόδου του αισθητήρα
line_valid	Είσοδος	1 - bit	Έγκυρη γραμμή δεδομένων
frame_valid	Είσοδος	1 - bit	Έγκυρο πλαίσιο δεδομένων
init_done	Είσοδος	1 - bit	Ολοκλήρωση αρχικοποίησης καταχωρητών μέσω του I <sup>2</sup> C πρωτοκόλλου
start	Έξοδος	1 - bit	Ενεργοποίηση I <sup>2</sup> C πρωτοκόλλου
fra_done	Είσοδος	1 - bit	Ολοκλήρωση λήψης πλαισίου
interrupt	Είσοδος	1 - bit	Ενεργοποίηση αναμονής
red_pixels	Έξοδος (inout)	24 - bits	Σύνολο της τιμής της κόκκινης χρωματικής συνιστώσας των pixel της εικόνας
green_pixels	Έξοδος (inout)	25 - bits	Σύνολο της τιμής της πράσινης χρωματικής συνιστώσας των pixel της εικόνας
blue_pixels	Έξοδος (inout)	24 - bits	Σύνολο της τιμής της μπλε χρωματικής συνιστώσας των pixel της εικόνας
rd_data	Έξοδος	16 - bits	Μεταφορά των pixel στο υπόλοιπο σύστημα ύστερα από αίτηση ανάγνωσης

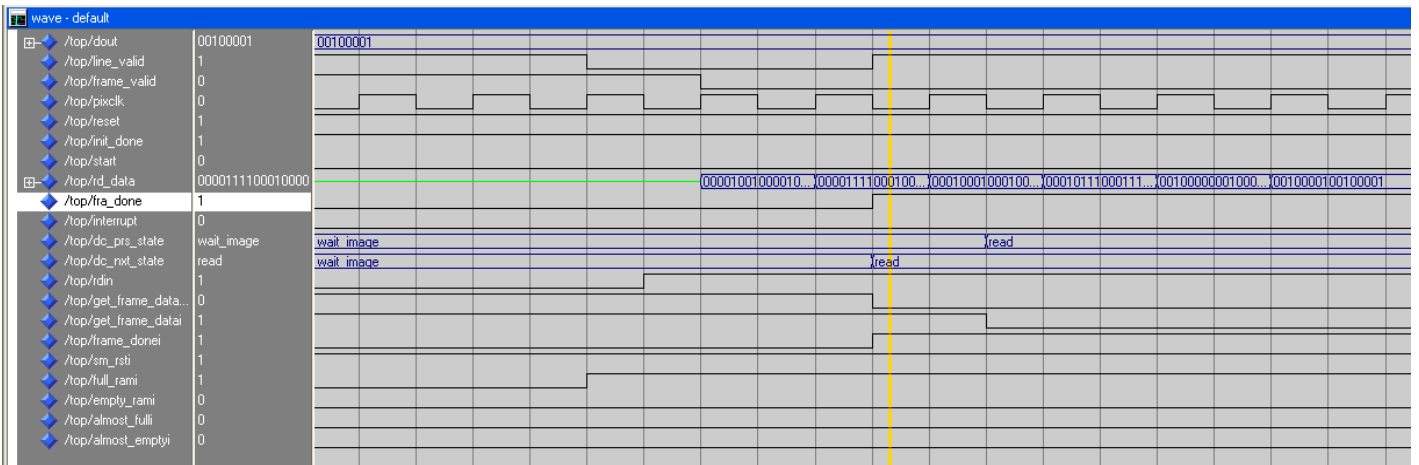
**Πίνακας 7.7:** Περιγραφή πυλών κυκλώματος ελέγχου συστήματος

Το κύκλωμα ελέγχθηκε για τη σωστή λειτουργία με τη χρήση του Modelsim. Τα αποτελέσματα της προσομοίωσης και της σωστής λειτουργίας φαίνονται στα επόμενα σχήματα. Στο σχήμα 19 φαίνεται η έναρξη λειτουργίας του κυκλώματος, γίνεται η αρχικοποίηση του I<sup>2</sup>C και αρχίζει η λήψη του πλαισίου. Στο σχήμα 20 παρατηρούμε ότι ολοκληρώνεται η λήψη του πλαισίου (frame\_done = 1) από τον αισθητήρα και αρχίζει η ανάγνωση της εικόνας από το σύστημα.



**Σχήμα 7.19:** Αρχικοποίηση I<sup>2</sup>C

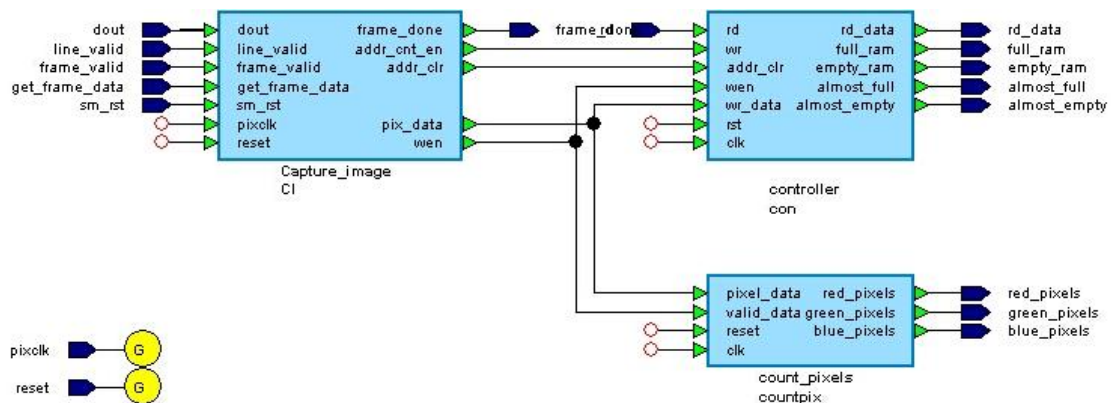
## Λειτουργική προσομοίωση



Σχήμα 7.20: Ολοκλήρωση λήψης εικόνας, ανάγνωση από το σύστημα

### 7.6.1 Κύκλωμα διασύνδεσης

Το κύκλωμα που συνδέει το σύστημα μνήμης με το κύκλωμα σύλληψης εικόνας απεικονίζεται στο επόμενο σχήμα.



Σχήμα 7.21: Κύκλωμα διασύνδεσης καταγραφής εικόνας και συστήματος μνήμης

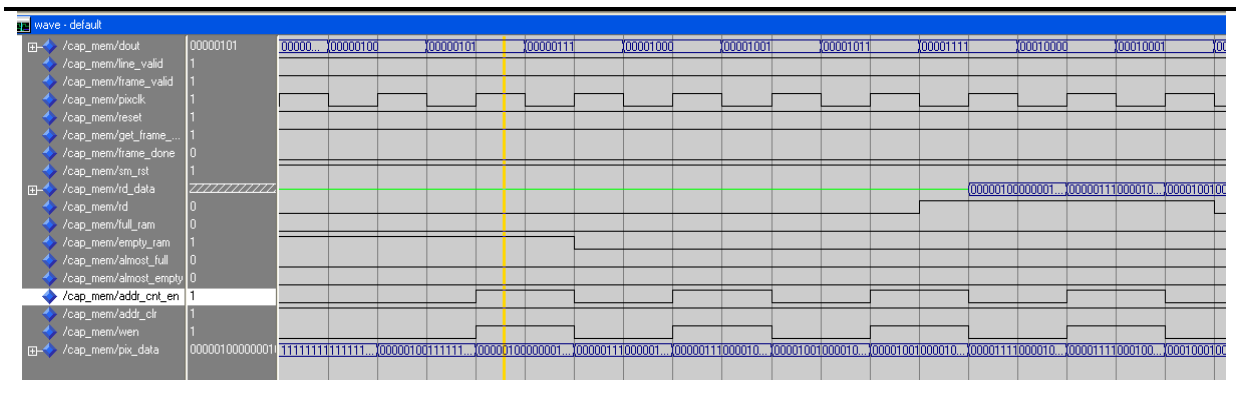
Στο επόμενο σχήμα απεικονίζεται στιγμιότυπο του κυκλώματος με χρήση του HDL DESIGNER με την περιγραφή των εισόδων και εξόδων.

## Λειτουργική προσομοίωση

A	B	C	D	E	F	G	H	I	J
1	Order	Name	Bounds	Type	cap_mem	cap_mem Capture_image controller	con	count_pixels	Comments
2		Instance Ref:				CI	con	countpi	
3		Frame Label:							
4		Port Map:				++	++	++	
5	1	addr_cnt_en		STD_LOGIC		o	I		
6	2	addr_clr		STD_LOGIC		o	I		
7	3	weni		STD_LOGIC		o	I	I	
8	4	pix_data	(15 downto 0)	STD_LOGIC_VECTOR		o	I	I	
9	5	dout	(7 downto 0)	std_logic_vector	I	I			
10	6	line_valid		STD_LOGIC	I	I			
11	7	frame_valid		STD_LOGIC	I	I			
12	8	pixclk		STD_LOGIC	I	I	I	I	
13	9	reset		STD_LOGIC	I	I	I	I	
14	10	get_frame_data		STD_LOGIC	I	I			
15	11	frame_done		STD_LOGIC	o	o			
16	12	sm_rst		STD_LOGIC	I	I			
17	13	rd_data	(15 downto 0)	std_logic_vector	o		o		
18	14	rd		std_logic	I		I		
19	15	full_ram		std_logic	o		o		
20	16	empty_ram		std_logic	o		o		
21	17	almost_full		std_logic	o		o		
22	18	almost_empty		std_logic	o		o		
23	19	red_pixels	(23 downto 0)	std_logic_vector	o			o	
24	20	green_pixels	(24 downto 0)	std_logic_vector	o			o	
25	21	blue_pixels	(23 downto 0)	std_logic_vector	o			o	
26									

**Πίνακας 7.8:** Περιγραφή πυλών κυκλώματος διασύνδεσης

Το κύκλωμα διασύνδεσης ελέγχθηκε για τη σωστή λειτουργία με τη χρήση του Modelsim. Τα αποτελέσματα της προσομοίωσης και της σωστής λειτουργίας φαίνονται στα επόμενα σχήματα. Στο σχήμα 22 φαίνεται η έναρξη λειτουργίας του κυκλώματος, όπου αρχίζει η εγγραφή της μνήμης ενώ γίνεται και ανάγνωση των πρώτων θέσεων της μνήμης παρατηρώντας ότι τα δεδομένα είναι τα ίδια στις αντίστοιχες θέσεις της μνήμης, αποδεικνύοντας τη σωστή λειτουργία του κυκλώματος.



**Σχήμα 7.22:** Εγγραφή και ανάγνωση εικόνας

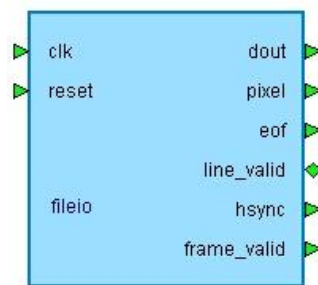
Στο σχήμα 23 φαίνεται η ενεργοποίηση του σήματος `almost_empty` δηλώνοντας ότι η μνήμη πρόκειται να αδειάσει και παρατηρούμε ότι όντως μετά από κάποιες περιόδους η μνήμη αδειάζει ενεργοποιώντας το σήμα `empty_ram='1'`.

ns	delta	/cap_mem/dout	/cap_mem/get_frame_data	/cap_mem/rd_data	/cap_mem/almost_empty	/cap_mem/pix_data
75950	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	0 1 1 1 0010000100100001
75950	+2	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76000	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76050	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76050	+2	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76100	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76150	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76200	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76250	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76300	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76350	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76400	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76450	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76500	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76550	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76600	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76650	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76700	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76750	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76800	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76850	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76900	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76950	+0	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 0 0	1 1 1 1 0010000100100001
76950	+3	00100001	0 0 1 1	0 1 1 0010000100100001	1 1 1 0	0 1 1 1 0010000100100001
77000	+0	00100001	0 0 0 1	0 1 1 0010000100100001	1 1 1 0	0 1 1 1 0010000100100001

Σχήμα 7.23: Ολοκλήρωση ανάγνωσης

## 7.7 Ανάγνωση εικόνας

Για τον έλεγχο της σωστής λειτουργίας του κυκλώματος και για να υποδυθούμε τη πραγματική λειτουργία του αισθητήρα γίνεται ανάγνωση μιας έγχρωμης εικόνας με τη χρήση του Matlab και στην συνέχεια ανάγνωση από ένα κύκλωμα το οποίο τροφοδοτεί με pixels τη διεπαφή παίρνοντας το ρόλο του αισθητήρα.

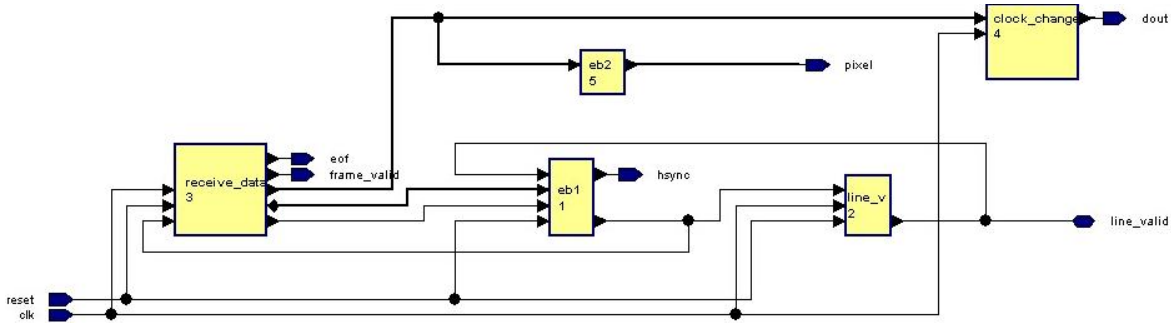


Σχήμα 7.24: Block διάγραμμα κυκλώματος ανάγνωσης εικόνας

Το κύκλωμα ανάγνωσης του αρχείου της εικόνας απεικονίζεται στο επόμενο σχήμα.



## Λειτουργική προσομοίωση



Σχήμα 7.25: Κύκλωμα ανάγνωσης εικόνας

Παρατηρούμε ότι το αποτέλεσμα της ανάγνωσης μιας έγχρωμης εικόνας με τη χρήση του Matlab είναι ένας πίνακας 3 διαστάσεων που περιέχει τις τιμές του κόκκινου, πράσινου και μπλε χρώματος κάθε pixel, χρησιμοποιώντας 8-bit για την αναπαράσταση της κάθε τιμής. Οπότε κάθε pixel είναι 24-bits. Επειδή ο αισθητήρας παράγει pixels των 16-bits γίνεται μετατροπή της RGB εικόνας σε μονοδιάστατο πίνακα χρησιμοποιώντας 16-bit για κάθε pixel. Στη συνέχεια αποθηκεύουμε τις τιμές κάθε pixel σε ένα αρχείο κειμένου το οποίο αποτελεί είσοδο του κυκλώματος ανάγνωσης.

Το κύκλωμα αυτό εκτός των άλλων, καθώς διαβάζει την εικόνα, αυξάνει ένα μετρητή στο τέλος της κάθε γραμμής του αρχείου μέχρι να φθάσει στο τέλος γραμμής του πλαισίου. Κάθε γραμμή του αρχείου της εικόνας είναι ουσιαστικά ένα pixel μεγέθους 16-bit. Μόλις φθάσουμε στο τέλος γραμμής το σήμα `line_valid` γίνεται 0, όπως θα έκανε και ο αισθητήρας, ενώ στο τέλος του αρχείου γίνεται μηδέν το σήμα `frame_valid` δηλώνοντας το τέλος του πλαισίου. Το `line_valid` ουσιαστικά αντιπροσωπεύει πληροφορία HSYNC ενώ το `frame_valid` πληροφορία VSYNC που είναι σημαντική για την απεικόνιση της εικόνας. Επιπλέον, επειδή ο αισθητήρας παράγει στην έξοδο 8-bit κάθε φορά και όχι 16-bit όσο είναι το μέγεθος ενός pixel, το παραπάνω κύκλωμα χωρίζει τα 16-bit που διαβάζει από το αρχείο σε 8-bit στέλνοντας, πρώτα τα σημαντικότερα 8-bit και στη συνέχεια τα υπόλοιπα όπως ακριβώς και ο αισθητήρας.

Το παραπάνω κύκλωμα υλοποιήθηκε σε vhd1 με χρήση του πακέτου txtio και ελέγχθηκε για τη σωστή λειτουργία με τη χρήση του Modelsim. Τα αποτελέσματα της προσομοίωσης και της σωστής λειτουργίας φαίνονται στο επόμενο σχήμα. Τα δεδομένα που διαβάζει αποθηκεύονται στο σήμα `pixel_int`.

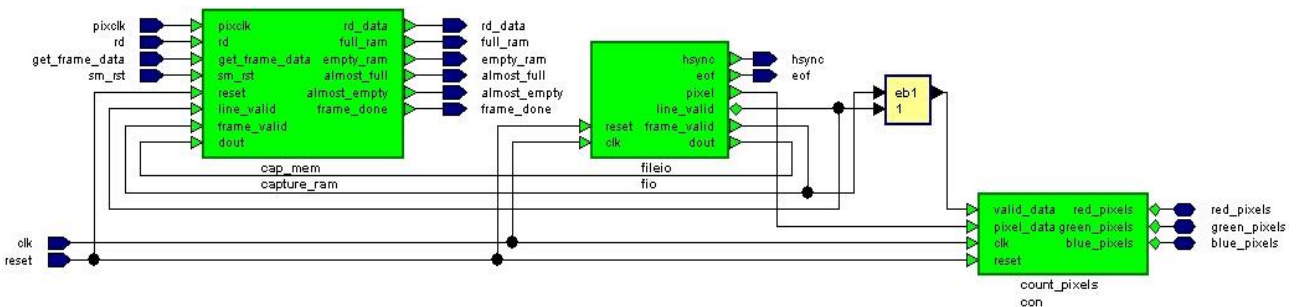
## Λειτουργική προσομοίωση



Σχήμα 7.26: Ανάγνωση εικόνας

## 7.8 Το συνολικό κύκλωμα

Παρακάτω απεικονίζεται το συνολικό κύκλωμα με διασυνδεδεμένα τα επιμέρους κυκλώματα, το κύκλωμα σύλληψης εικόνας και το σύστημα μνήμης, το κύκλωμα ανάγνωσης και το κύκλωμα μέτρησης των χρωματικών συνιστωσών..



Σχήμα 7.27: Συνολικό κύκλωμα

Τα αποτελέσματα της προσομοίωσης και της σωστής λειτουργίας φαίνονται στα επόμενα τα. Στο σχήμα 28 γίνεται έναρξη ανάγνωσης ενός πλαισίου και αποθήκευσης στη μνήμη. Παρατηρούμε το σήμα ram\_empty να είναι μηδέν δηλώνοντας ότι η μνήμη άρχισε να εγγράφεται. Επίσης παρατηρούμε τους μετρητές των χρωματικών συνιστωσών. Στο σχήμα 29 φαίνεται το τέλος πλαισίου και η ενεργοποίηση του σήματος eof το οποίο αντιστοιχεί σε πληροφορία vsync.

Όσον αφορά το ρολόι του αισθητήρα αυτό είναι ίσο με το μισό της συχνότητας του ρολογιού του συστήματος.



## Λειτουργική προσομοίωση

ns	delta	/rd_data	/final/almost_empty	/final/eof	/final/red_pixels	/final/green_pixels	/final/blue_pixels		
		/final/rd	/final/get_frame_data				/final/valid		
		/final/full_ram	/final/frame_done				/final/line_valid		
		/final/empty_ram	/final/sm_rst				/final/frame_valid		
		/final/almost_full	/final/hsync						
260925	+0	ZZZZZZZZZZ	0 0 0 0	0	1 0 1 0 0	000000001001010110101001	0000000010011110011010010	000000001001110011000001	1 1 1
260950	+0	ZZZZZZZZZZ	0 0 0 0	0	1 0 1 0 0	000000001001010110101001	0000000010011110011010010	000000001001110011000001	1 1 1
260950	+1	ZZZZZZZZZZ	0 0 0 0	0	1 0 1 0 0	000000001001010110101001	0000000010011110100000001	000000001001110011100000	1 0 1
260950	+2	ZZZZZZZZZZ	0 0 0 0	0	1 0 1 1 0	000000001001010110101001	0000000010011110100000001	000000001001110011100000	0 1 1
260975	+0	ZZZZZZZZZZ	0 0 0 0	0	1 0 1 1 0	000000001001010110101001	0000000010011110100000001	000000001001110011100000	0 1 1
261000	+0	ZZZZZZZZZZ	0 0 0 0	0	1 0 1 1 0	000000001001010110101001	0000000010011110100000001	000000001001110011100000	0 1 1

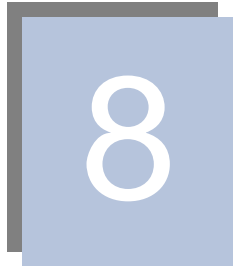
**Σχήμα 7.30:** Τέλος γραμμής πλαισίου

Τέλος στο σχήμα 31 φαίνεται η σωστή ανάγνωση των στοιχείων της μνήμης, το πρώτο στοιχείο είναι το 0000000000000000, το δεύτερο 0000000011011010 και τα λοιπά.

ns	delta	/final/pixclk	/final/rd_data	/final/almost_empty	/final/eof
		/final/clk	/final/rd	/final/get_frame_data	
		/final/reset	/final/full_ram	/final/frame_done	
			/final/empty_ram	/final/sm_rst	
			/final/almost_full	/final/hsync	
6507175	+1	1 1 1	0000000000000000	1 0 0 0	0 1 0 1 1 1
6507200	+0	0 0 1	0000000000000000	1 0 0 0	0 1 0 1 1 1
6507225	+0	1 0 1	0000000000000000	1 0 0 0	0 1 0 1 1 1
6507225	+1	1 0 1	0000000011011010	1 0 0 0	0 1 0 1 1 1
6507250	+0	0 1 1	0000000011011010	1 0 0 0	0 1 0 1 1 1
6507275	+0	1 1 1	0000000011011010	1 0 0 0	0 1 0 1 1 1
6507275	+1	1 1 1	0000000011011010	1 0 0 0	0 1 0 1 1 1
6507300	+0	0 0 1	0000000011011010	1 0 0 0	0 1 0 1 1 1
6507325	+0	1 0 1	0000000011011010	1 0 0 0	0 1 0 1 1 1
6507325	+1	1 0 1	0000001010001110	1 0 0 0	0 1 0 1 1 1
6507350	+0	0 1 1	0000001010001110	1 0 0 0	0 1 0 1 1 1
6507375	+0	1 1 1	0000001010001110	1 0 0 0	0 1 0 1 1 1
6507375	+1	1 1 1	0000001101101000	1 0 0 0	0 1 0 1 1 1
6507400	+0	0 0 1	0000001101101000	1 0 0 0	0 1 0 1 1 1
6507425	+0	1 0 1	0000001101101000	1 0 0 0	0 1 0 1 1 1
6507425	+1	1 0 1	0000010001000010	1 0 0 0	0 1 0 1 1 1
6507450	+0	0 1 1	0000010001000010	1 0 0 0	0 1 0 1 1 1
6507475	+0	1 1 1	0000010001000010	1 0 0 0	0 1 0 1 1 1
6507475	+1	1 1 1	0000010100011100	1 0 0 0	0 1 0 1 1 1
6507500	+0	0 0 1	0000010100011100	1 0 0 0	0 1 0 1 1 1
6507525	+0	1 0 1	0000010100011100	1 0 0 0	0 1 0 1 1 1
6507525	+1	1 0 1	0000010111110110	1 0 0 0	0 1 0 1 1 1
6507550	+0	0 1 1	0000010111110110	1 0 0 0	0 1 0 1 1 1
6507575	+0	1 1 1	0000010111110110	1 0 0 0	0 1 0 1 1 1
6507575	+1	1 1 1	0000011011010000	1 0 0 0	0 1 0 1 1 1
6507600	+0	0 0 1	0000011011010000	1 0 0 0	0 1 0 1 1 1
6507625	+0	1 0 1	0000011011010000	1 0 0 0	0 1 0 1 1 1
6507625	+1	1 0 1	000001110101010	1 0 0 0	0 1 0 1 1 1

**Σχήμα 7.31:** Ανάγνωση μνήμης

Από όλες τις παραπάνω προσομοιώσεις συμπεραίνουμε τη σωστή λειτουργία όλων των κυκλωμάτων. Αναφέρουμε τέλος ότι υλοποιήθηκε η συνάρτηση του λογαρίθμου ώστε να είναι δυνατός ο δυναμικός ορισμός των μεγεθών των αρτηριών. Ουσιαστικά επιστρέφει τον αριθμό των bits που χρειάζεται η αναπαράσταση ενός αριθμού όπως για παράδειγμα το μέγεθος της μνήμης.

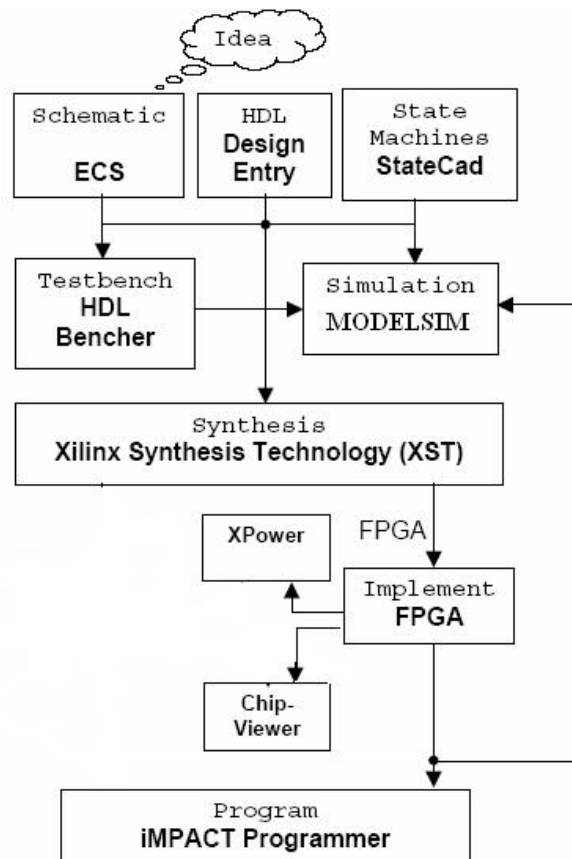


## Σύνθεση και Υλοποίηση

Εφόσον επιβεβαιωθεί η σωστή λειτουργία του κυκλώματος μέσω της λειτουργικής προσομοίωσης προχωρούμε στη σύνθεση του κυκλώματος και τη δημιουργία του αρχείου netlist, απαραίτητο για τα επόμενα στάδια της ροής πληροφορίας. Στο παρόν κεφάλαιο λοιπόν περιγράφεται η ροή πληροφορίας του εργαλείου που χρησιμοποιήσαμε για την ενσωμάτωση της σχεδίασης στο FPGA. Για κάθε κύκλωμα ξεχωριστά παρατίθενται τα αποτελέσματα της σύνθεσης, της μετάφρασης, του map και του Place And Route. Επιπλέον για κάθε κύκλωμα παρατίθεται εικονική αναπαράσταση της τοποθέτησής του στο FPGA. Λεπτομερείς αναφορές κάθε σταδίου παρατίθενται στο παράρτημα.

## 8.1 Ροή σχεδίασης - εργαλεία

Το στάδιο που ακολουθεί τη σχεδίαση της αρχιτεκτονικής και της περιγραφής της σε γλώσσα υλικού είναι η σύνθεση και η υλοποίηση του κυκλώματος. Για τα στάδια αυτά χρησιμοποιήθηκε το λογισμικό ISE 7.1i της εταιρείας Xilinx [40]. Το εργαλείο αυτό ελέγχει όλα τα στάδια της ροής σχεδίασης (design flow). Μέσω του Project Navigator έχουμε τη δυνατότητα χρήσης ποικίλων εργαλείων ολοκλήρωσης μιας σχεδίασης ενώ παρέχει τη δυνατότητα διασύνδεσης με άλλα προγράμματα όπως το εργαλείο Modelsim που χρησιμοποιήσαμε στην προσομοίωση των κυκλωμάτων και το Leonardo Spectrum που χρησιμοποιείται για τη σύνθεση μιας σχεδίασης. Υποστηρίζει τέλος σχεδόν όλες τις συσκευές FPGA της Xilinx. Στο επόμενο σχήμα περιγράφεται η ροή σχεδίασης και τα εργαλεία που χρησιμοποιούνται σε κάθε στάδιο.



Σχήμα 8.1: Ροή σχεδίασης εργαλείου ISE 7.1i

Η υλοποίηση για τα FPGA όπως περιγράψαμε και στο κεφάλαιο 5 απαιτεί τα επόμενα στάδια:

- **Translate :** γίνεται ερμηνεία της σχεδίασης και εκτελείται ο έλεγχος design rule (DRC)
- **MAP:** υπολογίζεται και τοποθετείται η σχεδίαση στις πηγές (recourses) της συσκευής
- **Place & Route:** οι CLB τοποθετούνται σε λογική θέση και χρησιμοποιούνται οι πόροι δρομολόγησης της συσκευής
- **Configuration:** δημιουργείται ένα bitstream αρχείο για την εγκατάσταση στο FPGA

Αρχικά λοιπόν δημιουργούμε ένα νέο project όπου τοποθετούμε τη σχεδίασή μας προσδιορίζοντας διάφορες παραμέτρους όπως τη συσκευή στόχο, τη γλώσσα περιγραφής υλικού και τον τύπο του προσομοιωτή που θα χρησιμοποιήσουμε [42]. Έχουμε τη δυνατότητα χρήσης σχηματικής ή vhdl περιγραφής της σχεδίασης. Υπάρχει επίσης η δυνατότητα σχεδίασης μηχανής πεπερασμένων καταστάσεων. Για να προχωρήσουμε στη λειτουργική προσομοίωση του κυκλώματος απαιτείται η χρήση αρχείων testbench όπου δίνονται τιμές στις εισόδους του κυκλώματος.

Αφού προσομοιώσουμε τη σχεδίαση στη συνέχεια ακολουθεί η σύνθεση του κυκλώματος όπου το κύκλωμα ελέγχεται για τυχόν λάθη και μετατρέπεται σε αρχείο NGC netlist ή EDIF. Σε αυτό το στάδιο ελέγχεται η σύνταξη και γίνεται μετάφραση και αντιστοίχιση τμημάτων του κώδικα σε ένα σύνολο από στοιχεία που το εργαλείο μπορεί να αναγνωρίζει. Σε αυτό το στάδιο έχουμε τη δυνατότητα να δούμε τη σχηματική περιγραφή του κυκλώματος και να φθάσουμε μέχρι και σε επίπεδο πυλών. Μετά την ολοκλήρωση προκύπτει μια αναφορά όπου υπάρχουν πληροφορίες σχετικά με το κύκλωμα όπως οι μηχανές πεπερασμένης κατάστασης, οι πόροι του FPGA που χρησιμοποιούνται. Στη συνέχεια δημιουργούμε κάποιο αρχείο περιορισμών οι οποίοι μπορεί να είναι χωρικοί ή χρονικοί και χρησιμοποιούνται για να θέσουν την απαιτούμενη συχνότητα ή να προσδιορίσουν τα απαιτούμενα pin-out της σχεδίασης. Ένας χρονικός περιορισμός επιβεβαιώνει ότι στοιχεία που ελέγχονται από το ρολόι όπως, Flip-flop, Ram, και Latch ικανοποιούν την περίοδο του ρολογιού που έχουμε θέσει.

Στη συνέχεια εκτελούμε τα στάδια της υλοποίησης δίνοντας σαν είσοδο το αρχείο NGC netlist ή EDIF. Αρχικά εκτελούμε τη μετάφραση (translate), όπου γίνεται μετατροπή των netlists και συγχώνευσή τους σε ένα μοναδικό netlist, το οποίο

περιγράφει τη λογική της σχεδίασης καθώς και οποιαδήποτε χρονικό ή χωρικό περιορισμό.

Το στάδιο MAP τοποθετεί λογικά στοιχεία του κυκλώματος στους CLB και IOB πόρους του FPGA. Εκτελεί βελτιστοποιήσεις και έλεγχο DRC στο τελικό mapped netlist.

Στις αναφορές που προκύπτουν μετά το τέλος των παραπάνω βημάτων περιέχονται τυχόν λάθη του σταδίου μετάφρασης ή πληροφορίες σχετικές με την χρήση των πόρων της συσκευής αντίστοιχα. Στη φάση αυτή μπορούμε να εκτελέσουμε post map static timing για να δούμε μια εκτίμηση των καθυστερήσεων και να προσδιορίσουμε εάν οι χρονικοί περιορισμοί που έχουμε θέσει μπορούν να επιτευχθούν.

Το στάδιο Place And Route (PAR) ασχολείται με τα διαμορφωμένα λογικά block (CLBS – configurable logic blocks) και επιλέγει τη καλύτερη θέση για κάθε block. Για ένα γρήγορο μονοπάτι έχει νόημα η τοποθέτηση σχετικών CLBS το ένα δίπλα στο άλλο για την μείωση του μήκους του μονοπατιού. Οι πόροι της δρομολόγησης τοποθετούνται στη συνέχεια σε κάθε σύνδεση, επιλέγοντας προσεκτικά τη καλύτερη δρομολόγηση. Για παράδειγμα, εάν ένα σήμα χρειάζεται σε πολλές περιοχές της σχεδίασης το εργαλείο Place And Route θα χρησιμοποιήσει μια μεγάλη γραμμή (longline) για να γεφυρώσει το chip με την ελάχιστη καθυστέρηση. Με την ολοκλήρωση του σταδίου προκύπτουν δύο αναφορές. Στην αναφορά Place And Route περιέχεται μια περίληψη των καθυστερήσεων και μπορούμε να προσδιορίσουμε αν η σχεδίαση δρομολογήθηκε σωστά και οι χρονικοί περιορισμοί έχουν ικανοποιηθεί. Στην αναφορά Pad αναφέρονται οι τοποθεσίες των pin που χρησιμοποιήθηκαν.

Στη συνέχεια μπορούμε να χρησιμοποιήσουμε το εργαλείο FPGA editor για να δούμε το κύκλωμα τοποθετημένο στο FPGA ενώ μπορούμε να κάνουμε αλλαγές στη σχεδίαση τοποθετώντας και δρομολογώντας κρίσιμα στοιχεία του κυκλώματος με το χέρι.

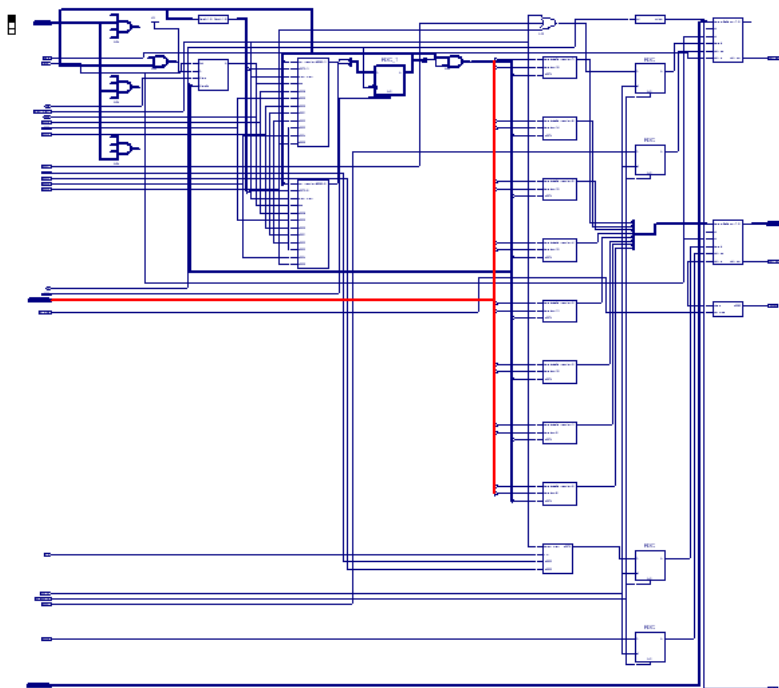
Τέλος μπορούμε να εκτελέσουμε Post-Layout Timing και να διαπιστώσουμε αν ικανοποιήθηκαν οι χρονικοί περιορισμοί. Αναφέρουμε ότι η πλειοψηφία των χειρότερων μονοπατιών αποτελείται από λογική καθυστέρηση. Γενικά μπορεί κανείς να μειώσει υπερβολικές καθυστερήσεις και να βελτιώσει την απόδοση μειώνοντας τον αριθμό των λογικών επιπέδων της σχεδίασης.



Τελικά ένα πρόγραμμα που ονομάζεται bitgen παίρνει την έξοδο του Place And Route και δημιουργεί ένα προγραμματισμένο bitstream (Configuration) μέσω του οποίου προγραμματίζεται το FPGA. Μπορεί όμως να μην είναι απαραίτητη η δημιουργία ενός bit αρχείου σε κάθε υλοποίηση αφού ο σχεδιαστής μπορεί να ενδιαφέρεται μόνο να διαβεβαιώσει ότι κάποιο συγκεκριμένο τμήμα της σχεδίασης περνά με επιτυχία τη προσομοίωση χρονισμού. Μετά το τέλος κάθε σταδίου υλοποίησης έχουμε τη δυνατότητα να εκτελέσουμε χρονική προσομοίωση. Επιπλέον μπορούμε να υπολογίσουμε την κατανάλωση ισχύος του κυκλώματος.

## 8.2 Πρωτόκολλο επικοινωνίας I<sup>2</sup>C

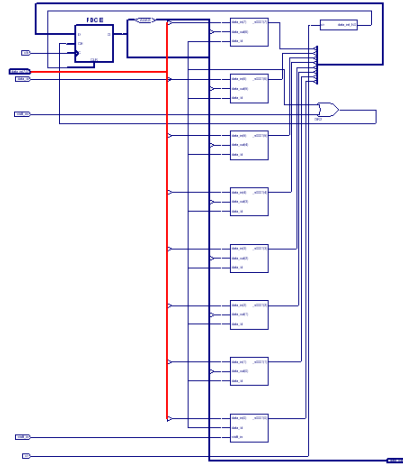
Εφόσον λοιπόν έχει επιβεβαιωθεί η σωστή λειτουργία του κυκλώματος μέσω της λειτουργικής προσομοίωσης ακολουθεί η σύνθεση του κυκλώματος και η δημιουργία του αρχείου netlist απαραίτητο για τα επόμενα στάδια της ροής πληροφορίας. Ελέγχουμε αρχικά τη σχεδίαση με την επιλογή check syntax και επιβεβαιώνουμε τη σωστή λειτουργία του κυκλώματος. Στο στάδιο της σύνθεσης έχουμε τη δυνατότητα να δούμε τη σχεδίαση σε σχηματικό διάγραμμα φθάνοντας ακόμα και σε επίπεδο πυλών.



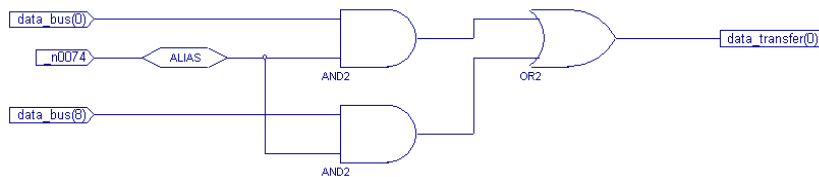
Σχήμα 8.2: Block διάγραμμα πρωτοκόλλου

## Σύνθεση και Υλοποίηση

Στο επόμενο σχήμα απεικονίζεται ο ολισθητής ενώ στο σχήμα 4 φθάνουμε σε επίπεδο πυλών, όπου παρατηρούμε τη δημιουργία ενός ψηφίου της αρτηρίας μεταφοράς δεδομένων.



Σχήμα 8.3: Ολισθητής κυκλώματος πρωτοκόλλου



Σχήμα 8.4: Επίπεδο πυλών

Στη συνέχεια φαίνεται η αναφορά της σύνθεσης, όπου μπορούμε να παρατηρήσουμε τα λογικά στοιχεία που χρησιμοποιήθηκαν, όπως flip flop και latches και την εκτίμηση των καθυστερήσεων. Λεπτομερής αναφορά σύνθεσης παρατίθεται στο παράρτημα.

---

Device utilization summary:

-----

Selected Device : 2s50eft256-7

Number of Slices:	73	out of	768	9%
Number of Slice Flip Flops:	71	out of	1536	4%
Number of 4 input LUTs:	129	out of	1536	8%
Number of bonded IOBs:	45	out of	182	24%
Number of GCLKs:	2	out of	4	50%

---

(a): Χρήση στοιχείων συσκευής στόχου

---

## Σύνθεση και Υλοποίηση

---

---

```
Minimum period: 10.532ns (Maximum Frequency: 94.949MHz)
Minimum input arrival time before clock: 6.532ns
Maximum output required time after clock: 9.946ns
Maximum combinational path delay: 9.352ns
```

---

---

(β): Αποτελέσματα χρονισμού

---

---

**Πίνακας 8.1:** Αναφορά σύνθεσης

Στη συνέχεια και εφόσον έχουμε ορίσει αρχείο χρονικών περιορισμών προχωρούμε στα στάδια υλοποίησης εκτελώντας translate, map, place and route προετοιμάζοντας το κύκλωμα για την μεταφορά στο FPGA.

---

---

```
Logic Utilization:
  Total Number Slice Registers:      58 out of  1,536   3%
    Number used as Flip Flops:              57
    Number used as Latches:                 1
  Number of 4 input LUTs:            129 out of  1,536   8%
Logic Distribution:
  Number of occupied Slices:                76 out of   768   9%
  Number of Slices containing only related logic:  76 out of    76 100%
  Number of Slices containing unrelated logic:    0 out of    76   0%
    *See NOTES below for an explanation of the effects of unrelated logic
Total Number 4 input LUTs:              133 out of  1,536   8%
  Number used as logic:                   129
  Number used as a route-thru:              4
  Number of bonded IOBs:                  44 out of   178  24%
  IOB Flip Flops:                          12
  IOB Latches:                              1
  Number of GCLKs:                          2 out of    4  50%
  Number of GCLKIOBs:                       1 out of    4  25%

Total equivalent gate count for design:  1,387
Additional JTAG gate count for IOBs:  2,160
Peak Memory Usage:  95 MB
```

---

---

**Πίνακας 8.2:** Αναφορά MAP

Στη παραπάνω αναφορά MAP παρατηρούμε ότι μόλις το 8% των συνολικών LUTs χρησιμοποιούνται ενώ ο αριθμός των πυλών για τη σχεδίαση είναι 1387.

Number of GCLKs	2 out of 4	50%
Number of External GCLKIOBs	1 out of 4	25%
Number of LOCed GCLKIOBs	0 out of 1	0%
Number of External IOBs	44 out of 178	24%
Number of LOCed IOBs	0 out of 44	0%
Number of SLICES	76 out of 768	9%

**Πίνακας 8.3:** Αναφορά Place And Route (χρήση στοιχείων)

## Σύνθεση και Υλοποίηση

---

Στην αναφορά Place And Route μπορούμε να δούμε τις πραγματικές καθυστερήσεις του κυκλώματος.

---

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

The AVERAGE CONNECTION DELAY for this design is: 1.023  
The MAXIMUM PIN DELAY IS: 3.309  
The AVERAGE CONNECTION DELAY on the 10 WORST NETS is: 2.476

---

**Πίνακας 8.4:** Αναφορά Place And Route (καθυστερήσεις (ns))

Μετά το τέλος της τοποθέτησης και δρομολόγησης της σχεδίασης εκτελούμε Post Place And Route Static Timing όπου φαίνονται τα αποτελέσματα χρονισμού και αναφέρεται αν ικανοποιήθηκαν οι περιορισμοί. Στο κύκλωμα θέσαμε 2 περιορισμούς όσον αφορά το ρολόι. Ο ένας αφορά τη συχνότητα εισόδου που τη θέσαμε στα 2MHz παρατηρώντας την ικανοποίησή του ενώ βλέπουμε ότι η ελάχιστη περίοδος είναι 6.863ns έχοντας έτσι τη δυνατότητα να αυξήσουμε κατά πολύ τη συχνότητα λειτουργίας της διεπαφής. Η περίοδος του περιορισμού είναι:

$$T_{\text{sys\_clk}} = \frac{1}{2 * 10^6 \text{ Hz}} = 500 \text{ nsec}$$

Ο άλλος περιορισμός είναι για το ρολόι scl που παράγει το πρωτόκολλο το οποίο θέσαμε στα 100KHz διαπιστώνοντας και πάλι την ικανοποίησή του.

$$T_{\text{scl}} = \frac{1}{100 * 10^3 \text{ Hz}} = 10 \text{ usec}$$

---

Timing constraint: TS\_sys\_clk = PERIOD TIMEGRP "sys\_clk" 500 ns HIGH 50%;

207 items analyzed, 0 timing errors detected. (0 setup errors, 0 hold errors)

Minimum period is 6.863ns.

---

Timing constraint: TS\_scl\_out\_reg1 = PERIOD TIMEGRP "scl\_out\_reg1" 10000 ns HIGH 50%;

288 items analyzed, 0 timing errors detected. (0 setup errors, 0 hold errors)

Minimum period is 11.692ns.

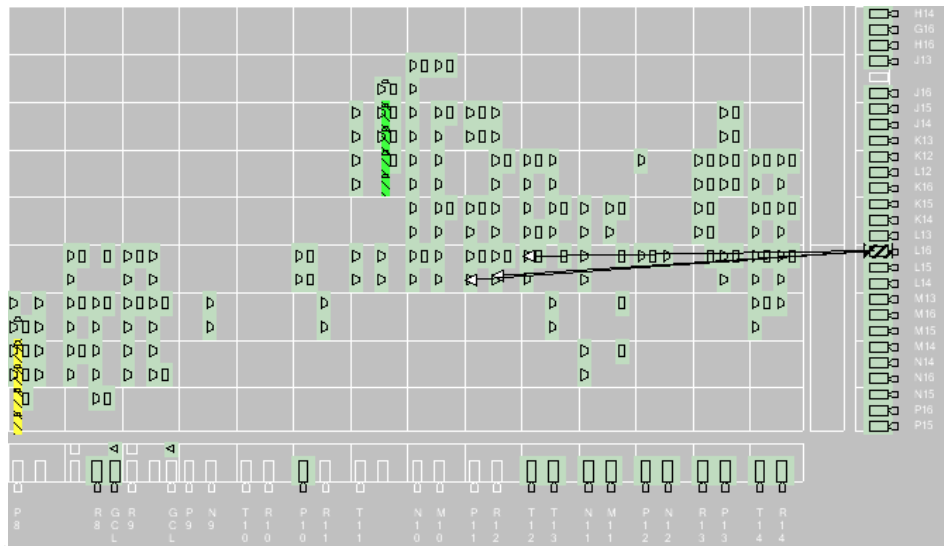
---

**Πίνακας 8.5:** Post Place And Route Static Timing

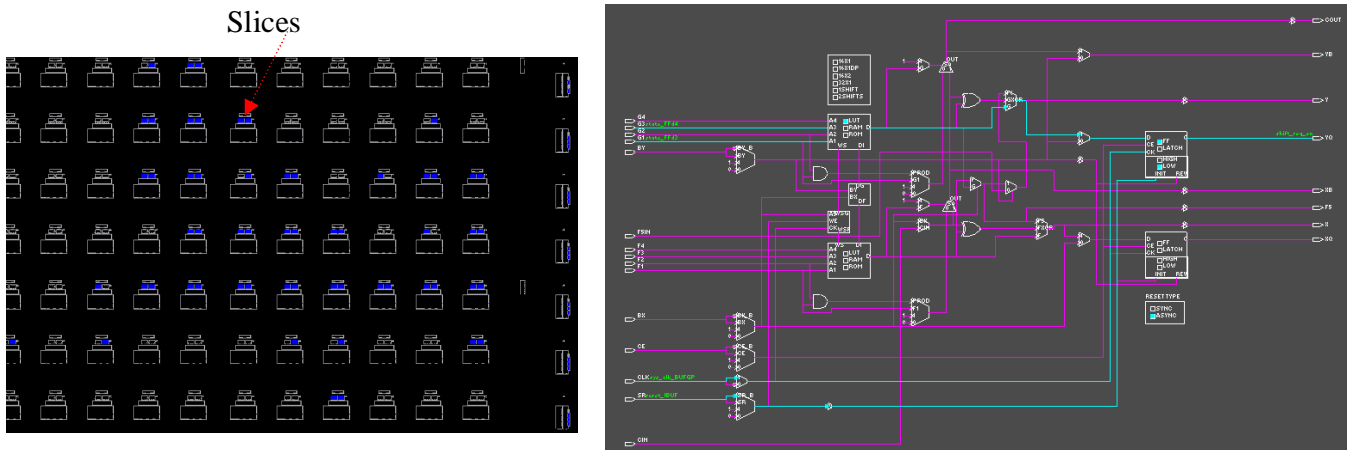
Επιπλέον μετά το τέλος της φάσης place and route μπορούμε να δούμε την τοποθέτηση του κυκλώματος στους πίνακες LUT του FPGA. Το σχήμα 5 αποτελεί στιγμιότυπο του εργαλείου floorplanner όπου μπορούμε να διακρίνουμε τα flip flop και

## Σύνθεση και Υλοποίηση

τους buffers που χρησιμοποιούνται. Το σχήμα 8α είναι αποτέλεσμα του εργαλείου FPGA editor όπου διακρίνουμε τα 2 slices από τα οποία αποτελείται κάθε λογική βαθμίδα σε κάθε ένα από τα οποία τοποθετείται ένα τμήμα της σχεδίασης το οποίο βλέπουμε στο σχήμα 8β όπου απεικονίζεται ένα στιγμιότυπο του ολισθητή της διεύθυνσης της slave συσκευή και το σήμα `shift_reg_en` από το οποίο και ελέγχεται.



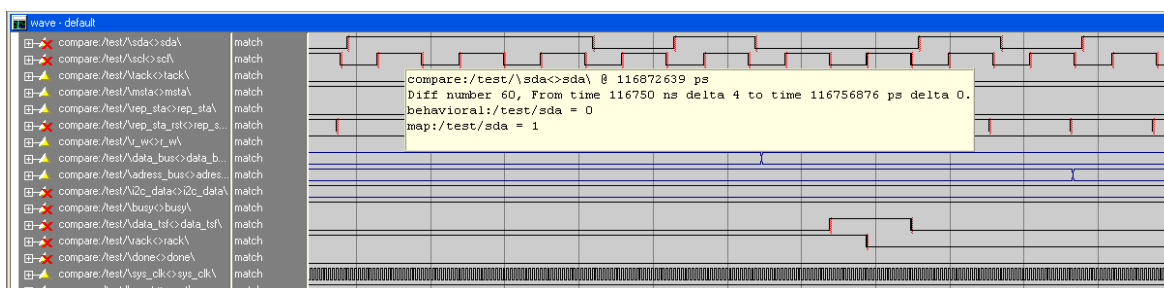
Σχήμα 8.5: Τοποθέτηση πρωτοκόλλου



Σχήμα 8.6: Τοποθέτηση κυκλώματος στο FPGA

(α): Τοποθέτηση κυκλώματος στο FPGA, (β): Τμήμα σχεδίασης σε ένα slice

Μετά την ολοκλήρωση κάθε σταδίου υλοποίησης εκτελούμε προσομοίωση του κυκλώματος για να διαπιστώσουμε την σωστή λειτουργία του κυκλώματος ακόμα και μετά την τοποθέτησή του στο FPGA. Στις προσομοιώσεις αυτές παρατηρούμε τις καθυστερήσεις στην παραγωγή της εξόδου που εισάγονται από την χρήση και προσομοίωση της τεχνολογίας 0.18/0.15 um στην οποία ανήκει το FPGA για το οποίο υλοποιήθηκε η σχεδίαση. Εκτελώντας σύγκριση των κυματομορφών που προκύπτουν από κάθε στάδιο διακρίνουμε τις καθυστερήσεις. Στο επόμενο σχήμα παρατηρούμε την καθυστέρηση στην παραγωγή του σήματος sda της τάξης κάποιων psec η οποία όμως δεν επηρεάζει τη λειτουργικότητα του κυκλώματος.



Σχήμα 8.7: Σύγκριση κυματομορφών λειτουργικής και Post Map προσομοίωσης

Παρακάτω απεικονίζονται τα αποτελέσματα από τον υπολογισμό κατανάλωσης ισχύος του κυκλώματος όπου παρατηρούμε ότι η συνολική ισχύς του κυκλώματος είναι 6.60 mW.

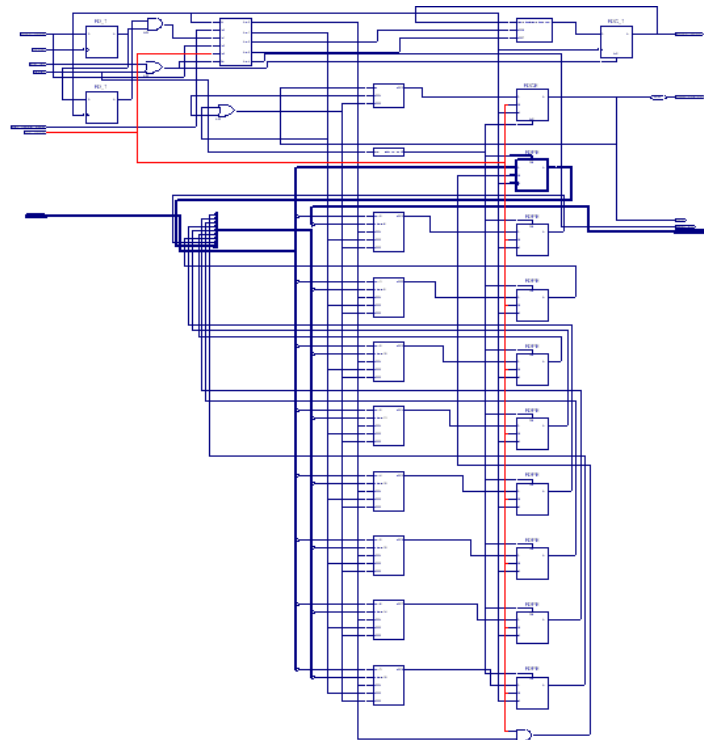
	Voltage [V]	Current [mA]	Power [mW]
<b>Vccint</b>	1.8		
Dynamic		0.00	0.00
Quiescent		0.00	0.00
<b>Vcco33</b>	3.3		
Dynamic		0.00	0.00
Quiescent		2.00	6.60
<b>Total Power</b>			6.60
Startup Current [m		500.00	
Battery Capacity [mA Hours]			0.00

Σχήμα 8.8: Κατανάλωση ισχύος

### 8.3 Καταγραφή εικόνας

Παρακάτω απεικονίζεται το αποτέλεσμα της σύνθεσης του κυκλώματος. Παρατηρούμε ότι κάθε διαδικασία του κυκλώματος που αναγνωρίζεται από το εργαλείο σύνθεσης τοποθετείται σε κατάλληλο block.

## Σύνθεση και Υλοποίηση



**Σχήμα 8.9:** Block διάγραμμα κυκλώματος καταγραφής εικόνας

Στους επόμενους πίνακες φαίνονται στιγμιότυπα της αναφοράς του σταδίου της σύνθεσης. Αρχικά παρατηρούμε την αναγνώριση της μηχανής πεπερασμένων καταστάσεων και ακολουθεί η αναφορά χρήσης των πόρων της συσκευής.

---

### Device utilization summary:

Selected Device : 2s50eft256-7

Number of Slices:	15	out of	768	1%
Number of Slice Flip Flops:	26	out of	1536	1%
Number of 4 input LUTs:	20	out of	1536	1%
Number of bonded IOBs:	34	out of	182	18%
Number of GCLKs:	1	out of	4	25%

---

**Πίνακας 8.6:** Χρήση πόρων συσκευής

Στον παραπάνω πίνακα εκτός από τα αποτελέσματα χρονοισμού απεικονίζεται ο υπολογισμός της καθυστέρησης ενός μονοπατιού. Στη συνέχεια εκτελούμε τα στάδια της υλοποίησης και παρακάτω απεικονίζονται συνοπτικά τα αποτελέσματα.

## Σύνθεση και Υλοποίηση

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops:	17	1,536	1%
Number of 4 input LUTs:	20	1,536	1%
Logic Distribution:			
Number of occupied Slices:	13	768	1%
Number of Slices containing only related logic:	13	13	100%
Number of Slices containing unrelated logic:	0	13	0%
Total Number of 4 input LUTs:	20	1,536	1%
Number of bonded IOBs:	33	178	18%
Number of GCLKs:	1	4	25%
Number of GCLKIOBs:	1	4	25%

**Σχήμα 8.10:** Συνοπτικά αποτελέσματα υλοποίησης σχεδίασης

Στη συνέχεια φαίνονται τα αποτελέσματα του Place And Route.

### Device Utilization Summary:

Number of GCLKs	1 out of 4	25%
Number of External GCLKIOBs	1 out of 4	25%
Number of LOCed GCLKIOBs	0 out of 1	0%
Number of External IOBs	33 out of 178	18%
Number of LOCed IOBs	0 out of 33	0%
Number of SLICES	13 out of 768	1%

### The Delay Summary Report

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

The AVERAGE CONNECTION DELAY for this design is:	1.216
The MAXIMUM PIN DELAY IS:	4.804
The AVERAGE CONNECTION DELAY on the 10 WORST NETS is:	2.113

**Πίνακας 8.7:** Αποτελέσματα Place And Route

Εκτελούμε Post Place And Route Static Timing διαπιστώνοντας την κανοποίηση των περιορισμού που έχουμε θέσει για τη συχνότητα του ρολογιού που ορίσαμε στα 13,5MHz. Παρατηρούμε ότι ο αισθητήρας μπορεί να λειτουργεί σε συχνότητα μέχρι 27MHz του ρολογιού εισόδου ενώ το ρολόι rixclk που παράγει ο αισθητήρας για την αποστολή των pixels τίθεται σε κάθε περίπτωση στη μισή συχνότητα του ρολογιού εισόδου. Έτσι λοιπόν θέσαμε τη συχνότητα εισόδου 27MHz όποτε το rixclk πρέπει να έχει 13,5MHz συχνότητα, δηλαδή περίοδο:

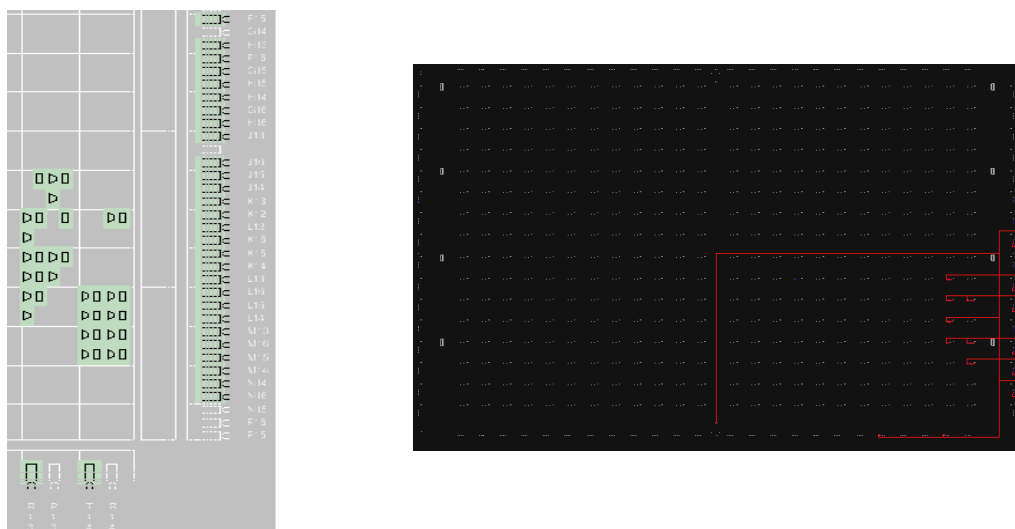
$$T = \frac{1}{13,5 * 10^6 \text{ Hz}} = 74074 \text{ psec}$$



Timing constraint: TS\_pixclk = PERIOD TIMEGRP "pixclk" 74.074 ns HIGH 50%;  
 54 items analyzed, 0 timing errors detected. (0 setup errors, 0 hold errors)  
 Minimum period is 9.528ns.

**Πίνακας 8.8:** Αποτελέσματα Post Place And Route Static Timing

Η τοποθέτηση του κυκλώματος φαίνεται στο επόμενο σχήμα. Με τη χρήση του FPGA editor μπορούμε να δούμε το κύκλωμα τοποθετημένο στο FPGA καθώς και κατανομή κάθε σήματος. Για παράδειγμα επιλέγουμε τα nets του κυκλώματος και από αυτά, το ρολόι pixclk και εμφανίζεται το fanout (=21) του δικτύου του ρολογιού.



**Σχήμα 8.11:** Τοποθέτηση κυκλώματος

(α) Τοποθέτηση κυκλώματος (εργαλείο Floorplanner), (β) Fanout ρολογιού

## 8.4 Σύστημα μνήμης

Στο επόμενο σχήμα απεικονίζεται το block διάγραμμα του κυκλώματος. Σε αυτό παρατηρούμε και τη χρήση αφαιρέτη που χρησιμοποιείται προαιρετικά στη σχεδίαση για τον προσδιορισμό των ελεύθερων θέσεων της μνήμης.



## Σύνθεση και Υλοποίηση

---

---

Selected Device : 2s50eft256-7

Number of Slices:	26	out of	768	3%
Number of Slice Flip Flops:	17	out of	1536	1%
Number of 4 input LUTs:	45	out of	1536	2%
Number of bonded IOBs:	42	out of	182	23%
Number of BRAMs:	1	out of	8	12%
Number of GCLKs:	1	out of	4	25%

---

(α): Χρήση στοιχείων συσκευής στόχου

---

Minimum period: 9.530ns (Maximum Frequency: 104.932MHz)  
Minimum input arrival time before clock: 7.614ns  
Maximum output required time after clock: 12.325ns  
Maximum combinational path delay: No path found

---

(β): Αποτελέσματα χρονισμού

---

**Πίνακας 8.9:** Αναφορά σύνθεσης

Παρατηρούμε ότι θέτοντας το μέγεθος της μνήμης 640 x 480 x 16 (το μέγιστο μέγεθος μιας εικόνας που παράγεται από τον αισθητήρα) η σύνθεση και υλοποίηση του κυκλώματος εξελίχθηκε δύσκολη καταναλώνοντας όλη την επεξεργαστική ισχύ και τους πόρους μνήμης του υπολογιστή όπου εκτελέστηκαν τα στάδια αυτά. Προκύπτει λοιπόν ότι απαιτείται η χρήση ξεχωριστής συσκευής μνήμης που θα συνδέεται στο FPGA ή μεγαλύτερου board που θα ενσωματώνει κυκλώματα μνήμης SRAM. Στη περίπτωση αυτή το παραπάνω στοιχείο μνήμης θα λειτουργεί σαν FIFO συγχρονίζοντας την αποστολή των pixel.

Ορίζουμε σαν περιορισμό την μέγιστη δυνατή συχνότητα του ρολογιού `pixclk` θέτοντας την περίοδο  $T=74074$  psec.

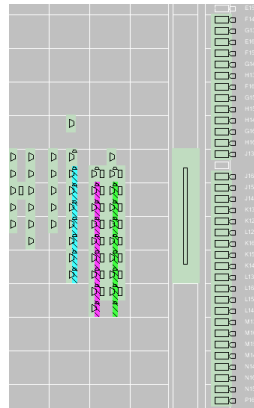
---

Timing constraint: TS\_clk = PERIOD TIMEGRP "clk" 74.074 ns HIGH 50%;  
1060 items analyzed, 0 timing errors detected. (0 setup errors, 0 hold errors)  
Minimum period is 8.453ns.

---

**Πίνακας 8.10:** Αποτελέσματα Post Static Timing

Στο επόμενο σχήμα παρατηρούμε τη τοποθέτηση του κυκλώματος ενώ διακρίνουμε τις αλυσίδες παραγωγής των ψηφίων των διευθύνσεων εγγραφής και ανάγνωσης.

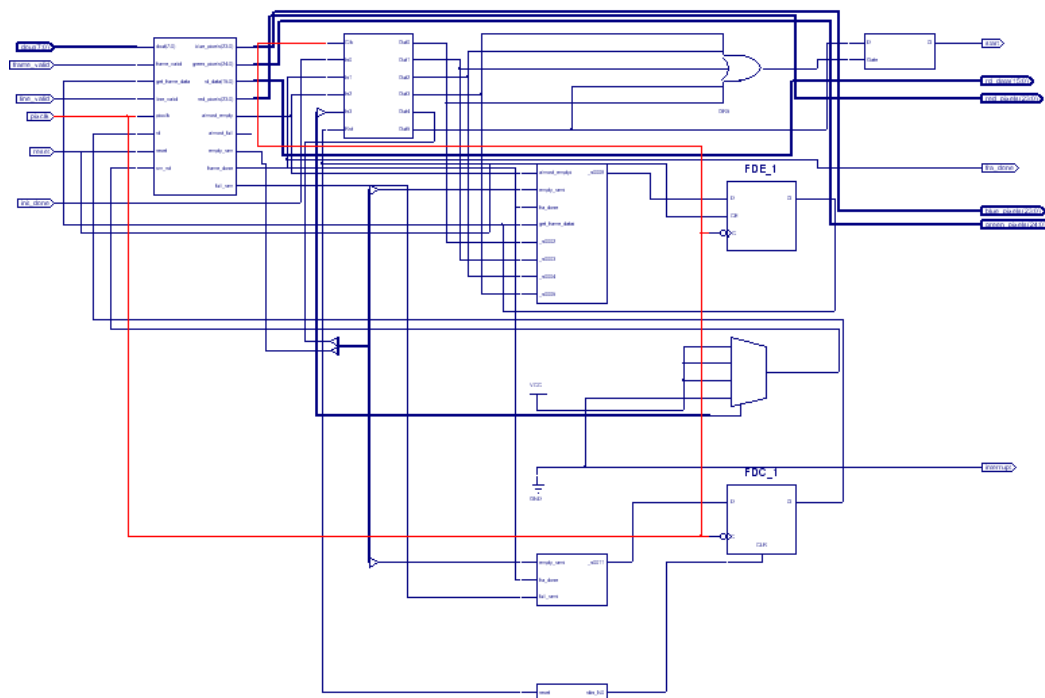


Σχήμα 8.14: Τοποθέτηση κυκλώματος στο FPGA (εργαλείο Floorplanner)

## 8.5 Συνολικό Κύκλωμα

### 8.5.1 Κύκλωμα ελέγχου συστήματος

Η σύνθεση του κυκλώματος ελέγχου που περιέχει και το κύκλωμα διασύνδεσης των κυκλωμάτων καταγραφής εικόνας, ελέγχου μνήμης και μέτρησης των χρωματικών συνιστωσών απεικονίζεται στο επόμενο σχήμα.



Σχήμα 8.15: Αποτελέσματα σύνθεσης - Block διάγραμμα κυκλώματος ελέγχου

## Σύνθεση και Υλοποίηση

Παρατηρούμε ότι τα αποτελέσματα χρονισμού της σύνθεσης αποτελούν μια εκτίμηση των καθυστερήσεων.

---

### Device utilization summary:

-----  
Selected Device : 2s50eft256-7

Number of Slices:	85	out of	768	11%
Number of Slice Flip Flops:	126	out of	1536	8%
Number of 4 input LUTs:	143	out of	1536	9%
Number of bonded IOBs:	105	out of	182	57%
Number of BRAMs:	1	out of	8	12%
Number of GCLKs:	1	out of	4	25%

---

(α): Χρήση στοιχείων συσκευής στόχου

---

Minimum period: 18.672ns (Maximum Frequency: 53.556MHz)  
Minimum input arrival time before clock: 5.851ns  
Maximum output required time after clock: 8.129ns  
Maximum combinational path delay: No path found

---

(β): Αποτελέσματα χρονισμού

---

### Πίνακας 8.11: Αναφορά σύνθεσης

---

#### Design Summary

-----  
Number of errors: 0  
Number of warnings: 2  
Logic Utilization:  
Number of Slice Flip Flops: 116 out of 1,536 7%  
Number of 4 input LUTs: 87 out of 1,536 5%  
Logic Distribution:  
Number of occupied Slices: 79 out of 768 10%  
Number of Slices containing only related logic: 79 out of 79 100%  
Number of Slices containing unrelated logic: 0 out of 79 0%  
\*See NOTES below for an explanation of the effects of unrelated logic  
Total Number 4 input LUTs: 145 out of 1,536 9%  
Number used as logic: 87  
Number used as a route-thru: 58  
Number of bonded IOBs: 104 out of 178 58%  
IOB Flip Flops: 9  
IOB Latches: 1  
Number of Block RAMs: 1 out of 8 12%  
Number of GCLKs: 1 out of 4 25%  
Number of GCLKIOBs: 1 out of 4 25%

Total equivalent gate count for design: 18,520  
Additional JTAG gate count for IOBs: 5,040  
Peak Memory Usage: 96 MB

---

### Πίνακας 8.12: Αναφορά MAP

### The Delay Summary Report

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

The AVERAGE CONNECTION DELAY for this design is: 1.316  
 The MAXIMUM PIN DELAY IS: 4.698  
 The AVERAGE CONNECTION DELAY on the 10 WORST NETS is: 4.003

### Device Utilization Summary:

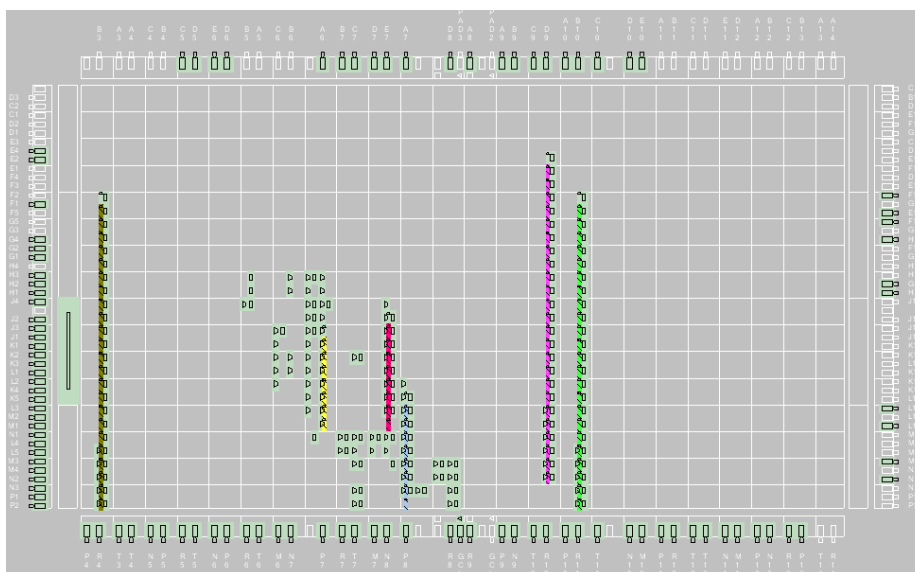
Number of BLOCKRAMs	1 out of 8	12%
Number of GCLKs	1 out of 4	25%
Number of External GCLKIOBs	1 out of 4	25%
Number of LOCed GCLKIOBs	0 out of 1	0%

Number of External IOBs	104 out of 178	58%
Number of LOCed IOBs	0 out of 104	0%

Number of SLICES	79 out of 768	10%
------------------	---------------	-----

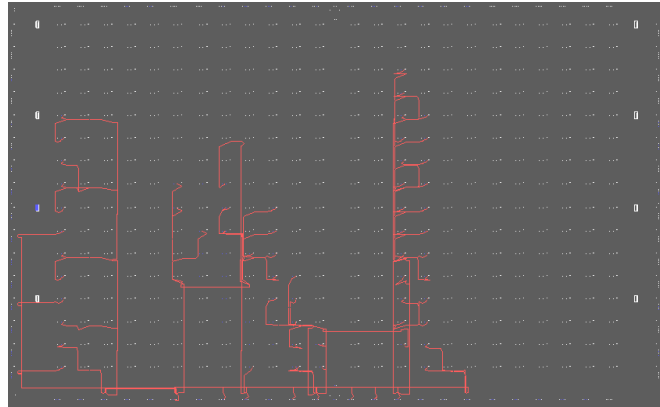
**Πίνακας 8.13:** Αναφορά PPlace And Route

Στο επόμενο σχήμα παρατηρούμε τα flip flop, τους buffers, τους πολυπλέκτες (στοιχείο δρομολόγησης) και τα άλλα λογικά στοιχεία (συστατικά μέρη ενός CLB) που χρησιμοποιούνται. Οι χρωματιστές γραμμές είναι αλυσίδες παραγωγής της εξόδου. Για παράδειγμα στα αριστερά και με το καφέ χρώμα είναι η αλυσίδα μέτρησης των χρωμάτων των pixel. Παρατηρούμε επίσης ότι τα σήματα που σχετίζονται τοποθετούνται σε μια γραμμή ώστε να μην προστίθεται πολυπλοκότητα αλλά και καθυστέρηση.



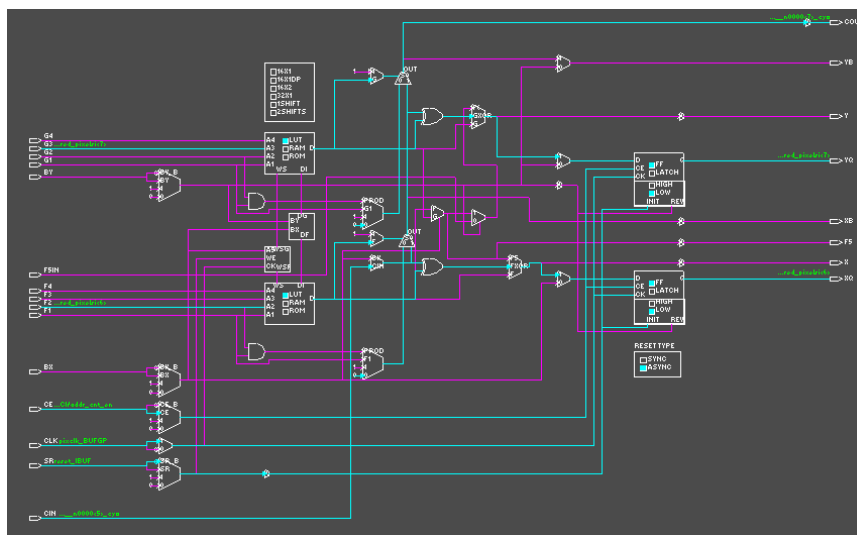
**Σχήμα 8.16:** Τοποθέτηση κυκλώματος (εργαλείο Floorplanner)

Στο επόμενο σχήμα φαίνεται η κατανομή του σήματος reset στο τοποθετημένο κύκλωμα πάνω στο FPGA.



Σχήμα 8.17: Δίκτυο σήματος reset

Όπως αναφέραμε μπορούμε να δούμε επίσης στοιχεία του κυκλώματος στη λεπτομερέστερη δυνατή μορφή εάν πατήσουμε σε κάποιο slice το οποίο αποθηκεύει στοιχεία του κυκλώματος σε λογικά κελιά όπως περιγράψαμε στο κεφάλαιο 5. Στο επόμενο σχήμα απεικονίζεται η παραγωγή ψηφίου της εξόδου του μετρητή `red_pixel`.



Σχήμα 8.18: Τοποθέτηση κυκλώματος στο FPGA (εργαλείο FPGA editor)

Θέτουμε και πάλι τη συχνότητα του ρολογιού στα 13,5MHz και εκτελούμε Post Place And Route Static Timing.

Timing constraint: TS\_pixclk = PERIOD TIMEGRP "pixclk" 74.074 ns HIGH 50%;  
 3858 items analyzed, 0 timing errors detected. (0 setup errors, 0 hold errors)  
 Minimum period is 17.216ns.

**Πίνακας 8.14:** Αποτελέσματα Post Place And Route Static Timing

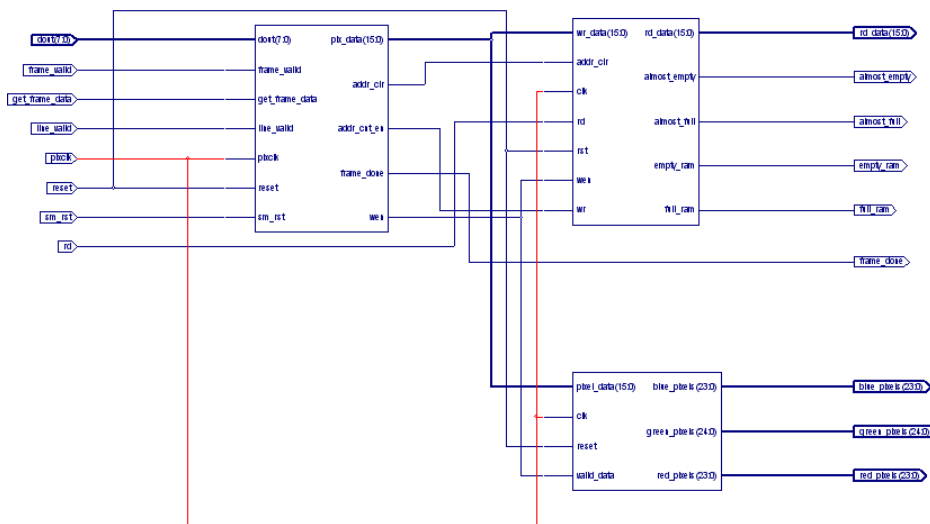
Παρακάτω απεικονίζονται τα αποτελέσματα από τον υπολογισμό κατανάλωσης ισχύος του κυκλώματος όπου παρατηρούμε ότι η συνολική ισχύς του συστήματος είναι 24.6mW.

	Voltage [V]	Current [mA]	Power [mW]
<b>Vccint</b>	1.8		
Dynamic		0.00	0.00
Quiescent		10.00	18.00
<b>Vcco33</b>	3.3		
Dynamic		0.00	0.00
Quiescent		2.00	6.60
<b>Total Power</b>			24.60
Startup Current [		500.00	

**Σχήμα 8.19:** Κατανάλωση ισχύος

### 8.5.2 Κύκλωμα διασύνδεσης

Παρακάτω απεικονίζεται η σύνθεση του κυκλώματος διασύνδεσης.

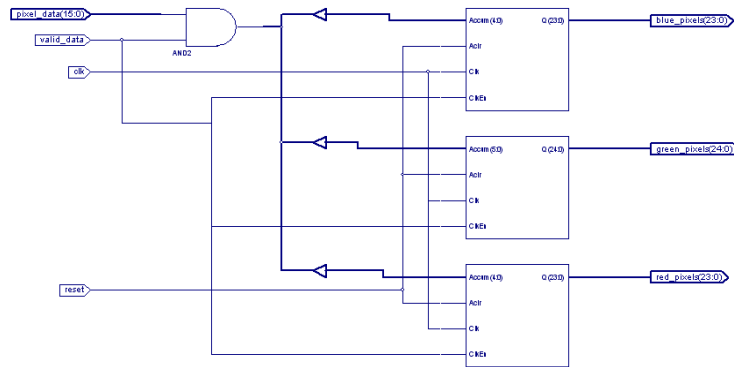


**Σχήμα 8.20:** Αποτελέσματα σύνθεσης - Block διάγραμμα κυκλώματος διασύνδεσης

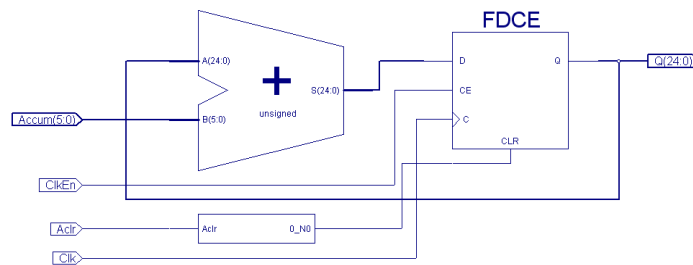


## Σύνθεση και Υλοποίηση

Στο επόμενο σχήμα απεικονίζεται το κύκλωμα μέτρησης των χρωματικών συνιστωσών και παρατηρούμε τους μετρητές που προκύπτουν από τη σύνθεση ο οποίος φαίνεται στο σχήμα 22.



Σχήμα 8.21: Μέτρηση χρωματικών συνιστωσών



Σχήμα 8.22: Μετρητής

### Device utilization summary:

Selected Device : 2s50eft256-7

Number of Slices:	72	out of	768	9%
Number of Slice Flip Flops:	117	out of	1536	7%
Number of 4 input LUTs:	130	out of	1536	8%
Number of bonded IOBs:	109	out of	182	59%
Number of BRAMs:	1	out of	8	12%
Number of GCLKs:	1	out of	4	25%

(α): Χρήση στοιχείων συσκευής στόχου

## Σύνθεση και Υλοποίηση

---

---

```
Minimum period: 12.022ns (Maximum Frequency: 83.181MHz)
Minimum input arrival time before clock: 5.851ns
Maximum output required time after clock: 12.170ns
Maximum combinational path delay: 6.383ns
```

---

(β): Αποτελέσματα χρονισμού

---

**Πίνακας 8.15:** Αναφορά σύνθεσης

---

---

Device Utilization Summary:

Number of BLOCKRAMs	1 out of 8	12%
Number of GCLKs	1 out of 4	25%
Number of External GCLKIOBs	1 out of 4	25%
Number of LOCed GCLKIOBs	0 out of 1	0%
Number of External IOBs	108 out of 178	60%
Number of LOCed IOBs	0 out of 108	0%
Number of SLICES	73 out of 768	9%

---

**Πίνακας 8.16:** Αναφορά PPlace And Route

Constraint	Requested	Actual	Logic Levels
TS_pixclk = PERIOD TIMEGRP "pixclk" 74.07	74.074ns	13.942ns	1

**Πίνακας 8.17:** Αποτελέσματα χρονισμού PPlace And Route

Εκτελούμε Post Place And Route Static Timing διαπιστώνοντας την ικανοποίηση των περιορισμού που έχουμε θέσει για τη συχνότητα του ρολογιού που ορίσαμε στα 13,5MHz (T=74074 psec). Στον πίνακα 18 παρατηρούμε ότι η σχεδίαση μπορεί να είναι αρκετά πιο γρήγορη.

---

```
Timing constraint: TS_pixclk = PERIOD TIMEGRP "pixclk" 74.074 ns HIGH 50%;
1072 items analyzed, 0 timing errors detected. (0 setup errors, 0 hold errors)
Minimum period is 12.174ns.
```

---

**Πίνακας 8.18:** Αποτελέσματα Post Place And Route Static Timing

## Σύνθεση και Υλοποίηση

Επιπλέον έχουμε τη δυνατότητα να παρατηρήσουμε τις καθυστερήσεις των συνδέσεων στο κύκλωμα. Ένα παράδειγμα φαίνεται στο επόμενο σχήμα όπου παρατηρούμε τον υπολογισμό της καθυστέρησης κάθε σύνδεσης που συμμετέχει στο μονοπάτι. Επιπλέον στο τέλος παρατηρούμε το ποσοστό προέλευσης, λογική και δρομολόγηση, της συνολικής καθυστέρησης του μονοπατιού.

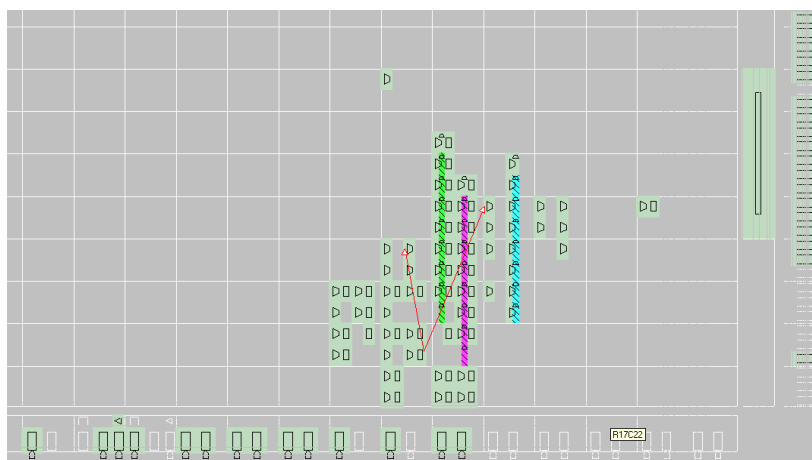
```
Delay: 10.293ns (data path - clock path skew + uncertainty)
Source: con/wri_adr_1 (FF)
Destination: con/rd_adr_6 (FF)
Data Path Delay: 10.293ns (Levels of Logic = 7)
Clock Path Skew: 0.000ns
Source Clock: pixclk_BUF0P rising
Destination Clock: pixclk_BUF0P rising
Clock Uncertainty: 0.000ns
```

Data Path: con/wri\_adr\_1 to con/rd\_adr\_6

Delay type	Delay(ns)	Logical Resource(s)
<u>Tcke</u>	0.886	con/wri_adr_1
net (fanout=6)	1.543	con/wri_adr<1>
<u>T0pcvg</u>	0.894	cap_mem_con/n0003<1>lut cap_mem_con/n0003<1>cv cap_mem_con/n0003<1>cv0
net (fanout=1)	0.000	cap_mem_con/n0003<2>cv cap_mem_con/n0003<2>cv0
<u>Tbvp</u>	0.133	cap_mem_con/n0003<3>cv cap_mem_con/n0003<3>cv0
net (fanout=1)	0.000	cap_mem_con/n0003<4>cv cap_mem_con/n0003<4>cv0
<u>Tbvp</u>	0.133	cap_mem_con/n0003<5>cv cap_mem_con/n0003<5>cv0
net (fanout=1)	0.000	cap_mem_con/n0003<6>xor cap_mem_con/n0003<6>xor
<u>Tcinx</u>	0.594	con/n0003<6>
net (fanout=1)	1.306	con/n000421
<u>Tilo</u>	0.418	CHOICE100
net (fanout=1)	1.083	con/n000430
<u>Tilo</u>	0.418	empty_ram_0BUF
net (fanout=3)	0.321	con/n00171
<u>Tilo</u>	0.418	con/n0017
net (fanout=5)	1.532	con/rd_adr_6
<u>Tceck</u>	0.614	con/rd_adr_6
Total	10.293ns	(4.508ns logic, 5.785ns route) (43.8% logic, 56.2% route)

Σχήμα 8.23: Καθυστέρηση μονοπατιού - Post Place And Route Static Timing

Τέλος στο επόμενο σχήμα παρατηρούμε τα flip flop, τους buffers και τα άλλα λογικά στοιχεία που χρησιμοποιούνται στο κύκλωμα αυτό.



Σχήμα 8.24: Τοποθέτηση κυκλώματος στο FPGA (εργαλείο Floorplanner)

Στον επόμενο πίνακα απεικονίζεται παράδειγμα αρχείου χρονικών και χωρικών περιορισμών για την ενσωμάτωση της σχεδίασης στο FPGA και τον προσδιορισμό της τοποθέτησης των εξόδων και εισόδων στο board που είχαμε στη διάθεσή μας.

```
NET "bt<0>" LOC = "P3" ;
NET "bt<1>" LOC = "P206" ;
NET "bt<2>" LOC = "P44" ;
NET "bt<3>" LOC = "P43" ;
NET "bt<4>" LOC = "P42" ;
NET "btn" LOC = "P187" ;
NET "ld<0>" LOC = "P111" ;
NET "ld<1>" LOC = "P109" ;
NET "ld<2>" LOC = "P102" ;
NET "ld<3>" LOC = "P100" ;
NET "ld<4>" LOC = "P98" ;
NET "ld<5>" LOC = "P96" ;
NET "ld<6>" LOC = "P94" ;
NET "ld<7>" LOC = "P89" ;
NET "ldg" LOC = "P45" ;
NET "led" LOC = "P154" ;
NET "pixclk" LOC = "P182" ;
NET "sw<0>" LOC = "P23" ;
NET "sw<1>" LOC = "P21" ;
NET "sw<2>" LOC = "P18" ;
NET "sw<3>" LOC = "P16" ;
NET "sw<4>" LOC = "P11" ;
NET "sw<5>" LOC = "P9" ;
NET "sw<6>" LOC = "P7" ;
NET "sw<7>" LOC = "P5" ;
NET "pixclk" TNM_NET = "pixclk";
TIMESPEC "TS_pixclk" = PERIOD "pixclk" 74074 ps HIGH 50 %;
```

**Πίνακας 8.19:** Αρχείο χρονικών και χωρικών περιορισμών



## Συμπεράσματα

### Άξονες Μελλοντικής Ανάπτυξης

---

#### 9.1 Συμπεράσματα

Στην διπλωματική αυτή λοιπόν υλοποιήθηκαν οι δύο πιο σημαντικοί αλγόριθμοι προ επεξεργασίας εικόνας οι οποίοι κατορθώνουν να βελτιώνουν την ποιότητά της ώστε να είναι διαθέσιμη στο υπόλοιπο σύστημα. Οι αλγόριθμοι αυτοί υλοποιήθηκαν στην πολύπλοκή και όχι στην απλή τους μορφή επιτρέποντας τη βελτίωση της εικόνας ακόμα και εάν δεν είναι γνωστός ο φωτισμός που χρησιμοποιήθηκε τη στιγμή της καταγραφής. Αυτό είναι απαραίτητο γιατί σήμερα με την ανάπτυξη των δικτύων τρίτης και τέταρτης γενιάς που επιτρέπουν την επικοινωνία μέσω εικονομυνημάτων θα πρέπει η ποιότητα της εικόνας που παράγεται να είναι βέλτιστη ώστε να γίνει συμπίεση και να μεταδοθεί σε κάποιο δίκτυο. Έτσι θα φθάσει στα χέρια μας μια εικόνα με την καλύτερη δυνατή απεικόνιση της πραγματικής σκηνής η οποία θα μπορεί να βρίσκεται χιλιόμετρα μακριά.

Υλοποιήθηκαν επίσης τα υποκυκλώματα εκείνα που συνιστούν την διεπαφή του συστήματος με τον αισθητήρα ακολουθώντας τη ροή πληροφορίας ενσωμάτωσης της σχεδίασης σε FPGA που παρουσιάστηκε στο 5<sup>ο</sup> κεφάλαιο. Στο 7<sup>ο</sup> κεφάλαιο παρουσιάστηκαν τα αποτελέσματα της λογικής προσομοίωσης των κυκλωμάτων όπου αποδεικνύεται η ορθή και επιθυμητή λειτουργία τους. Είναι σαφές ότι λειτουργούν με βάση την αρχιτεκτονική που παρουσιάστηκε στο κεφάλαιο 6.

Επιπλέον, σε αυτά ενσωματώνονται σήματα τα οποία μπορούν να χρησιμοποιηθούν από το μελλοντικό ολοκληρωμένο σύστημα για τον καλύτερο έλεγχο της καταγραφής εικόνας. Τέτοια σήματα είναι τα `almost_empty` και `almost_full` που σηματοδοτούν ότι η μνήμη πρόκειται να αδειάσει ή να γεμίσει. Επίσης, οι μετρητές των χρωματικών συνιστωσών διαθέτουν τις τιμές των χρωμάτων της εικόνας απευθείας στο

σύστημα και ιδιαίτερα στο λογισμικό, το οποίο ενσωματώνει και τους αλγορίθμους επεξεργασίας, υπολογίζοντας τα απαραίτητα στατιστικά της εικόνας χωρίς καθυστέρηση. Σημειώνουμε ότι δεν υλοποιήθηκε ο υπολογισμός των στατιστικών σε υλικό γιατί απαιτεί την εκτέλεση διαιρέσεων που απαιτούν μεγάλη επεξεργαστική ισχύ για την ολοκλήρωσή τους. Έτσι είναι προτιμότερη η εκτέλεσή τους από το λογισμικό.

Παρόμοια σήματα προστέθηκαν και στο κύκλωμα υλοποίησης του πρωτοκόλλου επικοινωνίας επιτρέποντας τον πιο γρήγορο προγραμματισμό των καταχωρητών του αισθητήρα. Έτσι αμέσως μόλις τελειώνει μια μεταφορά αλλά και ο προγραμματισμός ενός καταχωρητή ενεργοποιούνται τα σήματα `data_transfer` και `done`, αντίστοιχα, ώστε να σταλούν αμέσως τα δεδομένα για τον προγραμματισμό του επόμενου καταχωρητή.

Επιπρόσθετα ο ορισμός των μεγεθών που απαιτούνται στη σχεδίαση όπως το μέγεθος της μνήμης ή το μέγεθος των αρτηριών γίνεται δυναμικά επιτρέποντας στο χρήστη, αλλάζοντας τη τιμή μόνο μιας μεταβλητής στην αρχή της προσομοίωσης, αυτόματα να αλλάζουν όλα τα μεγέθη των σημάτων που εξαρτώνται από τη τιμή αυτή.

Ακολουθώντας τη ροή πληροφορίας στο κεφάλαιο 8 παρουσιάστηκαν τα αποτελέσματα της σύνθεσης και της υλοποίησης της σχεδίασης για κάθε κύκλωμα ξεχωριστά. Από αυτά συμπεραίνουμε την ορθή λειτουργία τους αλλά και ότι διατηρούν τη λειτουργικότητά τους ακόμα και μετά την εφαρμογή των σταδίων αυτών. Αυτό επιβεβαιώνεται και με την Post Map, Translate και Place And Route προσομοίωση που εκτελέσαμε για κάθε κύκλωμα και την εισαγωγή των καθυστερήσεων που επιβάλλονται από την τεχνολογία κατασκευής.

Οι χρονικοί περιορισμοί που θέσαμε ικανοποιήθηκαν ενώ τα αποτελέσματα αποδεικνύουν ότι τα κυκλώματα μπορούν να λειτουργήσουν σε πολύ μεγαλύτερη συχνότητα. Παρατηρούμε ότι οι χρονικοί περιορισμοί που θέσαμε είναι σύμφωνοι με τις απαιτήσεις του αισθητήρα ο οποίος μπορεί να δεχθεί ρολόι μέχρι 27MHz θέτοντας έτσι άνω όριο στη συχνότητα λειτουργίας.

Ακόμα, εκτελέσαμε Post Place And Route στατικό έλεγχο χρονισμού, τα αποτελέσματα του οποίου αποδεικνύουν την ικανοποίηση των περιορισμών. Επιπλέον, δείχνουν ότι η καθυστέρηση κάθε κυκλώματος είναι ικανοποιητική και αρκετά μικρότερη από αυτή που επιβάλλουν οι περιορισμοί που θέσαμε.

Τέλος, παράγοντας αρχείο bit προγραμματίσαμε το fpga που είχαμε στη διάθεσή μας αναθέτοντας τις εξόδους και εισόδους του στα pins του συμβατού περιφερειακού board (Digilant) που διαθέταμε αποδεικνύοντας την ορθή λειτουργία των κυκλωμάτων

## 9.2 Μελλοντική εργασία

Ωστόσο, παρόλο την υλοποίηση των παραπάνω κυκλωμάτων για την ολοκλήρωση του System On Chip απαιτείται η σχεδίαση και υλοποίηση και άλλων συστατικών μερών του συστήματος ώστε να χρησιμοποιηθεί για επεξεργασία πολυμέσων, κωδικοποίηση και συμπίεση (compression) εικόνας.

Καταρχήν θα πρέπει να υλοποιηθεί το λογισμικό τμήμα της διεπαφής. Στην παρούσα διπλωματική ολοκληρώθηκαν οι αλγόριθμοι προ επεξεργασίας. Για την ολοκλήρωση του λογισμικού όμως απαιτείται η ικανοποίηση των παρακάτω προδιαγραφών.

- Το λογισμικό θα πρέπει να επικοινωνεί με την διεπαφή και να αποστέλλει εντολές. Τέτοιες εντολές είναι το να ζητήσει μία καινούργια εικόνα από τον αισθητήρα (στην παρούσα διπλωματική υλοποιήθηκε σε υλικό), να προκαλέσει δηλαδή την καταγραφή νέου πλαισίου, να στείλει νέες τιμές για auto exposure και white balance, αλγόριθμοι που υλοποιήθηκαν ήδη στην παρούσα διπλωματική και άλλα.
- Το λογισμικό θα πρέπει να διαβάσει τις τιμές των χρωμάτων και να τις χρησιμοποιεί για να υπολογίσει το auto exposure value, και gain values για την επόμενη εικόνα. Η μεταφορά των μέσων τιμών από το SIF στον επεξεργαστή PPC (σχήμα E.1) μπορεί να γίνει είτε μία φορά ανά εικόνα (στο τέλος της εικόνας), είτε σταδιακά σε διάφορες φάσεις κατά την διάρκεια του διαβάσματος της εικόνας.
- Το λογισμικό θα πρέπει να διαβάσει τα pixels RGB από τη μνήμη (FIFO) και να τα αποθηκεύει σε περιφερειακή μνήμη σε κάποιο format. Τα pixels αυτά θα αποστέλλονται σε κάποιο LCD driver και θα απεικονίζονται σε ένα LCD display.

Επίσης, θα πρέπει να χρησιμοποιηθεί ένα μεγαλύτερο FPGA όπως Virtex II Pro

το οποίο θα επιτρέπει την χρήση του ενσωματωμένου επεξεργαστή για την εκτέλεση των αλγορίθμων επεξεργασίας και του λογισμικού. Η αρχιτεκτονική για την ενσωμάτωση ενός συστήματος πολυμέσων στη Virtex II Pro απεικονίζεται στην εισαγωγή, στο σχήμα E.1.

Ακόμη, για την ολοκλήρωση του συστήματος και τη δυνατότητα συμπίεσης πρέπει να υλοποιηθούν κυκλώματα τα οποία θα εφαρμόζουν κάποιο αλγόριθμο συμπίεσης εικόνας όπως το κλασικό standard συμπίεσης εικόνας JPEG ή το νέο standard JPEG2000 το οποίο προσφέρει αρκετά πλεονεκτήματα και έχει ήδη αρχίσει να χρησιμοποιείται σε ολοένα και περισσότερα συστήματα. Επιπλέον, θα μπορούσε να υλοποιηθεί σύστημα που θα εφαρμόζει συμπίεση video υλοποιώντας κάποιον αλγόριθμο συμπίεσης video όπως το σύγχρονο standard συμπίεσης MPEG4. Επίσης θα πρέπει να υλοποιηθούν και κυκλώματα μετατροπής χρωματικών χώρων (color space) για τη χρήση του πλέον κατάλληλου ανάλογα με την εφαρμογή.

Τελικά το σύστημα θα μπορούσε να αποτελέσει τμήμα ενός κινητού τηλεφώνου ή μιας ψηφιακής κάμερας ενσωματώνοντας πολλά χαρακτηριστικά της σύγχρονης ψηφιακής τεχνολογίας.

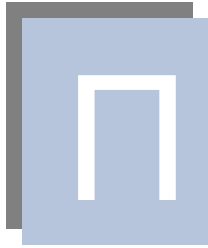
Εξάλλου αναμένεται η ανάπτυξη του ψηφιακού οπτικού κόσμου να είναι ραγδαία. Οι αισθητήρες CMOS βελτιώνονται συνεχώς και χρησιμοποιούνται ολοένα και περισσότερο σε περισσότερες εφαρμογές. Συστήματα με βιομετρικές εφαρμογές ενσωματώνουν πλέον συστήματα με δυνατότητα καταγραφής εικόνας αλλά και συστήματα παροχής οδηγικής βοήθειας αναπτύσσονται και ενσωματώνονται σε ολοένα και περισσότερα αυτοκίνητα τα οποία αναμένεται να κατακλύσουν την αυτοκινητοβιομηχανία στο μέλλον και να αποτελούν βασικά στοιχεία ανταγωνισμού και υπεροχής.

Η ανάπτυξη και εδραίωση των δικτύων τρίτης και τέταρτης γενιάς θα δημιουργήσει ανάγκη για ανάπτυξη νέων βελτιωμένων κινητών τηλεφώνων και ψηφιακών καμερών που θα πρέπει να ανταποκρίνονται στη ζήτηση και απαιτήσεις των καταναλωτών οι οποίοι επιθυμούν μικρότερα μεγέθη και καλύτερη ποιότητα εικόνας. Αφού η εικόνα στο μέλλον θα αποτελεί το βασικότερο μέσο επικοινωνίας, υποστηριζόμενη και από τη χωρητικότητα των σύγχρονων δικτύων, θα πρέπει η ποιότητά



της να είναι βέλτιστη και να επιτρέπει την καλύτερη δυνατή αναπαράσταση μιας σκηνης που θα βρίσκεται εκατοντάδες χιλιόμετρα μακριά.

Το σύστημά μας λοιπόν θα είναι εκεί για να ανταποκριθεί στις σύγχρονες αυτές ανάγκες.



## Παράρτημα

Σε αυτό το κεφάλαιο παρατίθενται περισσότερα αποτελέσματα των διαδικασιών σύνθεσης και υλοποίησης.

### Πρωτόκολλο επικοινωνίας I<sup>2</sup>C

Αναφορά της σύνθεσης όπου μπορούμε να παρατηρήσουμε τα λογικά στοιχεία που χρησιμοποιήθηκαν όπως flip flop και latches και εκτίμηση του καθυστερήσεων.

<b>Design Statistics</b>	<b>Cell Usage :</b>
# IOs : 45	# BELS : 152
	# GND : 1
	# INV : 4
<b>Macro Statistics :</b>	# LUT2 : 23
	# LUT2_D : 2
# FSMs : 2	# LUT3 : 32
# Adders/Subtractors : 1	# LUT3_D : 1
2-bit adder : 1	# LUT3_L : 3
	# LUT4 : 42
# Registers : 22	# LUT4_D : 3
# 1-bit register : 18	# LUT4_L : 23
# 2-bit register : 1	# MUXCY : 8
# 8-bit register : 3	# MUXF5 : 1
# Counters : 2	# VCC : 1
# 4-bit up counter : 2	# XORCY : 8
	<b># FlipFlops/Latches : 71</b>
	# FDC : 12
	# FDC_1 : 13
	# FDCE : 27
	# FDCPE : 8
	# FDP : 7
	# FDP_1 : 1
	# FDPE : 1
	# LDCE_1 : 2
	<b># Clock Buffers : 2</b>
	# BUFG : 1
	# BUFGP : 1
	<b># IO Buffers : 44</b>
	# IBUF : 29
	# OBUF : 15

**Πίνακας Π.1:** Λεπτομερής αναφορά σύνθεσης

## Καταγραφή εικόνας

Στους επόμενους πίνακες φαίνονται στιγμιότυπα της αναφοράς του σταδίου της σύνθεσης. Αρχικά παρατηρούμε την αναγνώριση της μηχανής πεπερασμένων καταστάσεων και ακολουθεί η αναφορά χρήσης των πόρων της συσκευής και τα αποτελέσματα χρονοισμού τα οποία αποτελούν μια εκτίμηση των καθυστερήσεων.

```
-----
| States                | 6
| Transitions           | 12
| Inputs                | 4
| Outputs               | 5
| Clock                 | pixclk (falling_edge)
| Reset                 | $n0003 (positive)
| Reset type            | asynchronous
| Reset State           | idle
| Power Up State        | idle
| Encoding               | automatic
| Implementation        | LUT
-----
Found 1-bit register for signal <addr_cnt_en>.
Found 1-bit register for signal <frame_done_int>.
Found 1-bit register for signal <frame_high>.
Found 1-bit register for signal <frame_low>.
Found 16-bit register for signal <pix_data_int>.
Summary:
inferred 1 Finite State Machine(s).
inferred 20 D-type flip-flop(s).
```

**Πίνακας Π.2:** Σύνθεση της μηχανής πεπερασμένων καταστάσεων

## Παράρτημα

---

```
Design Statistics
# IOs : 34

Macro Statistics :
# Registers : 20
# 1-bit register : 20

Cell Usage :
# BELS : 23
# GND : 1
# INV : 2
# LUT2 : 4
# LUT3 : 4
# LUT3_L : 1
# LUT4 : 3
# LUT4_L : 8
# FlipFlops/Latches : 26
# FD_1 : 2
# FDC_1 : 6
# FDCE : 1
# FDPE : 16
# FDPE_1 : 1
# Clock Buffers : 1
# BUFGP : 1
# IO Buffers : 33
# IBUF : 13
# OBUF : 20
=====
```

---

**Πίνακας Π.3:** Αποτελέσματα σύνθεσης

---

```
Design Summary
-----
Number of errors: 0
Number of warnings: 0
Logic Utilization:
  Number of Slice Flip Flops: 17 out of 1,536 1%
  Number of 4 input LUTs: 20 out of 1,536 1%
Logic Distribution:
  Number of occupied Slices: 13 out of 768 1%
  Number of Slices containing only related logic: 13 out of 13 100%
  Number of Slices containing unrelated logic: 0 out of 13 0%
  *See NOTES below for an explanation of the effects of unrelated logic
Total Number of 4 input LUTs: 20 out of 1,536 1%
  Number of bonded IOBs: 33 out of 178 18%
  IOB Flip Flops: 9
  Number of GCLKs: 1 out of 4 25%
  Number of GCLKIOBs: 1 out of 4 25%

Total equivalent gate count for design: 328
Additional JTAG gate count for IOBs: 1,632
Peak Memory Usage: 94 MB
```

---

**Πίνακας Π.4:** Αποτελέσματα MAP

**Σύστημα μνήμης**

<p>Design Statistics</p> <p># IOs : 42</p> <p>Macro Statistics :</p> <p># RAM : 1</p> <p># 256x16-bit dual-port block RAM : 1</p> <p># Registers : 1</p> <p># 1-bit register : 1</p> <p># Counters : 2</p> <p># 8-bit up counter : 2</p> <p># Tristates : 1</p> <p># 16-bit tristate buffer : 1</p> <p># Adders/Subtractors : 1</p> <p># 8-bit subtractor : 1</p>	<p>Cell Usage :</p> <p># BELS : 95</p> <p># GND : 1</p> <p># INV : 1</p> <p># LUT1 : 2</p> <p># LUT2 : 5</p> <p># LUT2_L : 8</p> <p># LUT3 : 19</p> <p># LUT3_L : 1</p> <p># LUT4 : 7</p> <p># LUT4_D : 1</p> <p># LUT4_L : 2</p> <p># MUXCY : 23</p> <p># VCC : 1</p> <p># XORCY : 24</p>
	<p># FlipFlops/Latches : 17</p> <p># FDCPE : 16</p> <p># FDP : 1</p> <p># RAMS : 1</p> <p># RAMB4_S16_S16 : 1</p> <p># Clock Buffers : 1</p> <p># BUFGP : 1</p> <p># IO Buffers : 41</p> <p># IBUF : 21</p> <p># OBUF : 4</p> <p># OBUFT : 16</p>

**Πίνακας Π.5:** Λεπτομερής αναφορά σύνθεσης

## Παράρτημα

---

---

```
Design Summary
-----
Number of errors:      0
Number of warnings:   2
Logic Utilization:
  Number of Slice Flip Flops:      17 out of 1,536   1%
  Number of 4 input LUTs:         43 out of 1,536   2%
Logic Distribution:
  Number of occupied Slices:                25 out of 768   3%
  Number of Slices containing only related logic:  25 out of 25 100%
  Number of Slices containing unrelated logic:    0 out of 25  0%
  *See NOTES below for an explanation of the effects of unrelated logic
Total Number 4 input LUTs:      45 out of 1,536   2%
  Number used as logic:                43
  Number used as a route-thru:         2
  Number of bonded IOBs:              41 out of 178 23%
  Number of Block RAMs:               1 out of 8   12%
  Number of GCLKs:                    1 out of 4   25%
  Number of GCLKIOBs:                 1 out of 4   25%

Total equivalent gate count for design: 16,967
Additional JTAG gate count for IOBs: 2,016
Peak Memory Usage: 95 MB
```

---

---

**Πίνακας Π.6:** Αναφορά MAP

---

---

### Device Utilization Summary:

Number of BLOCKRAMs	1 out of 8	12%
Number of GCLKs	1 out of 4	25%
Number of External GCLKIOBs	1 out of 4	25%
Number of LOCed GCLKIOBs	0 out of 1	0%
Number of External IOBs	41 out of 178	23%
Number of LOCed IOBs	0 out of 41	0%
Number of SLICES	25 out of 768	3%

### The Delay Summary Report

The AVERAGE CONNECTION DELAY for this design is:	1.255
The MAXIMUM PIN DELAY IS:	4.478
The AVERAGE CONNECTION DELAY on the 10 WORST NETS is:	3.733

---

---

**Πίνακας Π.7:** Αποτελέσματα Place And Route

**Συνολικό Κύκλωμα**

<p><b>Design Statistics</b></p> <p># IOs : 105</p> <p><b>Macro Statistics :</b></p> <p># RAM : 1 # 256x16-bit dual-port block RAM : 1</p> <p># Registers : 26 # 1-bit register : 23 # 25-bit register : 3</p> <p># Counters : 2 # 8-bit up counter : 2</p> <p># Multiplexers : 1 # 1-bit 4-to-1 multiplexer : 1</p> <p># Tristates : 1 # 16-bit tristate buffer : 1</p> <p># Adders/Subtractors : 4 # 25-bit adder : 3 # 8-bit subtractor : 1</p>	<p><b>Cell Usage :</b></p> <p># BELS : 335 # GND : 1 # INV : 3 # LUT1 : 57 # LUT2 : 8 # LUT2_D : 2 # LUT2_L : 22 # LUT3 : 26 # LUT4 : 21 # LUT4_L : 7 # MUXCY : 93 # VCC : 1 # XORCY : 94</p>
	<p># FlipFlops/Latches : 126 # FD_1 : 2 # FDC : 1 # FDC_1 : 13 # FDCE : 73 # FDCPE : 16 # FDE_1 : 1 # FDP : 1 # FDP_1 : 1 # FDPE : 16 # FDPE_1 : 1 # LD : 1 # RAMS : 1 # RAMB4_S16_S16 : 1 # Clock Buffers : 1 # BUFGP : 1 # IO Buffers : 104 # IBUF : 12 # OBUF : 76 # OBUFT : 16</p>

**Πίνακας Π.8:** Αναφορά σύνθεσης



## Βιβλιογραφία

---

- [1]. Fundamentals of Digital Image Processing, Anil K. Jain.
- [2]. Ψηφιακή επεξεργασία εικόνας, Ιωάννης Πήτας.
- [3]. The IMAGE PROCESSING Handbook, *Third Edition*, John C. Russ.
- [4]. Digital Image Processing, 5<sup>th</sup> edition, Bernd Jahne.
- [5]. VHDL for Designers, Stefan Sjöholm and Lennart Lindh.
- [6]. VHDL, Programming by Example, Fourth Edition, Douglas L. Perry.
- [7]. The VHDL Cookbook, Peter J. Ashenden, First Edition, 1990.
- [8]. "Color Image Processing Pipeline in Digital Still Cameras", IEEE Signal Processing , Rajeev Ramanath, Wesley E. Snyder, Youngjun F. Yoo, and Mark S. Drew, Special Issue on Color Image Processing, vol.22, no.1, Jan.2005, pp.34-43.
- [9]. J.E. Adams, K. Parluski, and K. Spaulding, "Color processing in digital cameras," *IEEE Micro*, vol. 18, no. 6, pp. 20–29, 1998.
- [10]. K. Parulski and K. Spaulding, "Color image processing for digital cameras," in *Digital Color Imaging Handbook*, G. Sharma, Ed. Boca Raton, FL: CRC Press, 2003, pp. 727–757.
- [11]. K. Illgner, H.G. Gruber, P. Gelabert, J. Liang, Y. Yoo, W. Rabadi, and R. Talluri, "Programmable DSC platform for digital still camera," in *Proc. IEEE ICASSP*, 1999, pp. 2235–2238.
- [12]. An Introduction to the Digital Still Camera Technology, Massimo Mancuso, Sebastiano Battiato.
- [13]. G. Wyszecki and W.S. Stiles, *Color Science—Concepts and Methods, Quantative Data and Formulae*, 2nd ed. New York: Wiley, 1982.
- [14]. S. S'usstrunk, R. Buckley, and S. Swen, "Standard RGB color spaces," in *Proc. IS T/SID 7th Color Imaging Conf.*, 1999, pp. 127–134.



- [15]. Standard RGB Color Spaces. Sabine Süsstrunk Laboratory of audio-visual Communication (EPFL), Lausanne, Switzerland. Robert Buckley Xerox Architecture Center, Webster, New York, USA. Steve Swen Apple Computer, Cupertino, California, USA.
- [16]. An Advanced Video Camera System with Robust AF, AE, and AWB Control June-Sok Lee, You-Young Jung, Byung-Soo Kim and Sung-Jea Ko, Senior Member, IEEE.
- [17]. Y. C. Cheng, W. H. Chan, and Y.Q. Chen, “Auto-matic White Balance for Digital Still Camera,” *IEEE Transactions on Consumer Electronics*, Volume 41, pp. 460-466, 1995.
- [18]. N. Kehtarnavaz, H.J. Oh, and Y. Yoo, “Development and real-time implementation of auto white balancing scoring algorithm,” *J. Real Time Imaging*, vol. 8, no. 5, pp. 379–386, 2002.
- [19]. Yoon Kim, June Sok Lee, Aldo W. Morales, Senior Member, IEEE Sung Jea Ko Senior Member, IEEE, “A video Camera System with enhanced zoom tracking and auto white balance”, *IEEE Transactions on Consumer Electronics*, Volume 48, No3 , August, 2002.
- [20]. K. Barnard, V. Cardei, and B. Funt, “A comparison of computational color constancy algorithms—Part I: Methodology and experiments with synthesized data,” *IEEE Trans Image Processing*, vol. 11, no. 9, pp. 972–983, 2002.
- [21]. G.D. Finlayson, S.D. Hordley, and I. Tastl, “Gamut constrained illuminant estimation,” in *Proc. ICCV03*, 2003, pp. 792–799.
- [22]. AUTOMATIC WHITE BALANCE FOR DIGITAL STILL CAMERA, Tzan-Sheng Chiou ,Chiou-Shann Fuh and Varsha Chikane Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.
- [23]. Preferred Color Spaces for White Balancing, Feng Xiao, Joyce E. Farrellb, Jeffrey M. DiCarloc and Brian A. Wandell.
- [24]. A new automatic exposure system for digital still cameras. Kuno,T. 1998. *IEEE transactions on consumer Electronics*, Vol 44, No 1. p 192-199.
- [25]. System Implications of Implementing Auto-Exposure on Consumer Digital Cameras, Robert Kremens, Nitin Sampat, Shyam Venkataraman and Thomas Yeh.

- [26]. S. Shimizu et al., "A New Algorithm for Exposure Control Based on Fuzzy Logic for Video Cameras," *IEEE Trans. Consumer Electronics*, vol. 38, no. 3, pp. 617–623, Aug. 1992.
- [27]. P. M. Hubel, J. Holm, G. D. Finlayson, "Illuminant Estimation and Color Correction," *Proc. Colour Imaging in Multimedia – CIM98*, Derby, England, pp. 97-105, 1998.
- [28]. System Implications of Implementing Auto-Exposure on Consumer Digital Cameras Robert Kremens, Nitin Sampat, Shyam Venkataraman and Thomas Yeh
- [29]. A. Morimura, K. Uomori, Y. Kitamura, A. Fujioka, J. Harada, S. Iwanura, and M. Hirota, "A digital video camera system," *IEEE Trans. Consumer Electron.*, vol. 36, no. 4, pp. 866–875, 1990.
- [30]. B.K. Gunturk, Y. Altunbasak, and R.M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Processing*, vol. 11, no. 9, pp. 485–492, 2002.
- [31]. B.K. Gunturk, J. Glotzbach, Y. Altunbasak, R.M. Mersereau, and R.W. Schafer, "Demosaicking: Color filter array interpolation," *IEEE Signal Processing Mag.*, vol. 22, no. 1, pp. 44–54, 2005.
- [32]. M. Subbarao and J.K. Tyan, "Selecting the optimal focus measure for autofocussing and depth-from-focus," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 8, pp. 864–870, 1998.
- [33]. J. Lee, K. Kim, and B. Nam, "Implementation of passive automatic focusing algorithm for digital still camera," *IEEE Trans. Consumer Electron.*, vol. 41, no. 3, pp. 449–454, 1995.
- [34]. DIGITAL CAMERA SYSTEM ON A CHIP, Eric R. Fossum, *IEEE Micro*, June 1998.
- [35]. Single-Chip CMOS Image Sensor for Mobile Applications, Kwangho Yoon, Chanki Kim, Bumha Lee, and Doyoung Lee, *IEEE Journal OF SOLID-STATE CIRCUITS*, vol. 37, no. 12, December 2002.
- [36]. MT9V112 SOC VGA DIGITAL IMAGE SENSOR, Specifications, Micron.
- [38]. THE I2C-BUS SPECIFICATION, VERSION 2.1, JANUARY 2000.

- [37]. Programmable Logic Design. Quick Start HandBook. Karan Parnell, Nick Mehta, Xilinx, 2003.
- [38]. Modelsim SE Tutorial.
- [39]. HDL Designer Series Tutorial.
- [40]. ISE 7 In-Depth Tutorial.
- [41]. ISE Quick Start Tutorial.
- [42]. Digilent DIO4 Peripheral Board Reference Manual.
- [43]. Digilent D2-SB System Board Reference Manual.

### **Ηλεκτρονική Βιβλιογραφία**

- [44]. Image Processing Fundamentals:  
<http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Contents.html>
- [45]. Micron Image sensors:  
<http://www.micron.com/products/imaging/technology/pixel.html>
- [46]. Image Sensors:  
<http://www.shortcourses.com/how/sensors/sensors.htm>
- [47]. VLSI design flow:  
<http://lsiwww.epfl.ch/LSI2001/teaching/webcourse/flow/flow.html>
- [48]. I2C (Inter-Integrated Circuit) Bus Technical Overview:  
<http://www.esacademy.com/faq/i2c>