



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη Πλήρως Ψηφιακού Πομπού Αναλογικού
Σήματος Εικόνας στα VHF σε FPGA**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΩΝΣΤΑΝΤΙΝΟΥ Ι. ΒΑΣΙΛΕΙΟΥ

Επιβλέπων : Παύλος - Πέτρος Σωτηριάδης
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2013



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη Πλήρως Ψηφιακού Πομπού Αναλογικού
Σήματος Εικόνας στα VHF σε FPGA**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΩΝΣΤΑΝΤΙΝΟΥ Ι. ΒΑΣΙΛΕΙΟΥ

Επιβλέπων : Πάυλος - Πέτρος Σωτηριάδης
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Πάυλος - Πέτρος Σωτηριάδης
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Ιωάννης Παπανάνος
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Κιαμάλ Πεκμεστζή
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2013

(Υπογραφή)

.....
ΚΩΝΣΤΑΝΤΙΝΟΥ Ι. ΒΑΣΙΛΕΙΟΥ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2013 – All rights reserved

Copyright © Κωνσταντίνος Ι. Βασιλείου, 2013

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.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Κίνητρο

Ο σκοπός της διπλωματικής εργασίας ήταν η σχεδίαση και υλοποίηση ενός πλήρως ψηφιακού πομπού αναλογικού σήματος εικόνας στη περιοχή συχνοτήτων εκπομπής της αναλογικής τηλεόρασης . Ο λόγος που έγινε αυτό είναι για να επιδεχθεί η δυνατότητα υλοποίησης αναλογικών κυκλωμάτων με αμιγώς ψηφιακές τεχνολογίες. Τα ψηφιακά κυκλώματα έχουν να επιδείξουν πολλά πλεονεκτήματα σε σχέση με τα αναλογικά. Ενώ για κάθε εφαρμογή το αναλογικό κύκλωμα απαιτεί λεπτομερή και εξειδικευμένο σχεδιασμό, η σχεδίαση του ψηφιακού μέρους γίνεται αυτόματα και εύκολα με τη χρήση εργαλείων αυτοματοποίησης σχεδίασης (Electronic Design Automation, EDA Tools), προσφέροντας ευκολία και μειώνοντας το χρόνο σχεδίασης και επομένως το χρόνο που απαιτείται για να κυκλοφορήσει ένα προϊόν στην αγορά. Η συνεχώς αυξανόμενη τάση για μείωση της τάσης τροφοδοσίας των κυκλωμάτων δημιουργεί προβλήματα στη λειτουργία των αναλογικών κυκλωμάτων, κάτι που δεν εμφανίζεται στα αντίστοιχα ψηφιακά. Επιπλέον τα αναλογικά και τα μικτά αναλογικά/ψηφιακά κυκλώματα απαιτούν τη εξειδικευμένων κυκλωμάτων (Digital to Analog Converters, Time to Digital Converters, Voltage Controlled Oscillators κ.α.) Αφαιρώντας αυτά τα υποκυκλώματα η σχεδίαση γίνεται πιο εύκολη και λιγότερο περίπλοκη. Τέλος η ενίσχυση ενός πλήρως ψηφιακού κυκλώματος όπως αυτό που σχεδιάστηκε και υλοποιήθηκε κατά τη διάρκεια αυτής της διπλωματικής εργασίας, γίνεται εύκολα και αποδοτικά με τη χρήση διακοπών (κλάσης D) ενισχυτών.

Περίληψη

Στα πλαίσια της παρούσας διπλωματικής εργασίας σχεδιάστηκε και υλοποιήθηκε σε τεχνολογία FPGA ένας πλήρως ψηφιακός πομπός αναλογικού σήματος εικόνας. Η περιγραφή του κυκλώματος έγινε στη γλώσσα Verilog, ενώ χρησιμοποιήθηκε FPGA της εταιρίας Xilinx.

Τα τεχνικά χαρακτηριστικά του πομπού περιλαμβάνουν: Χρήση VGA κάμερας με απαθανάτιση εικόνας διατάσεων 640x480 και ρυθμό ανανέωσης 30 fps, υλοποίηση αλγορίθμων για κωδικοποίηση ασπρόμαυρου σήματος εικόνας PAL (Phase Alternating Line), διαμόρφωση πλάτους και μέθοδο μετατροπής ψηφιακού σήματος σε αναλογικό με τη χρήση πηγής λευκού θορύβου.

Κατά τη διαδικασία της σχεδίασης λήφθηκε ιδιαίτερα υπόψη η σχεδίαση όσο πιο απλού κυκλώματος, έτσι ώστε να καταλαμβάνει τον ελάχιστο δυνατό χώρο στην επιφάνεια του FPGA και η δυνατότητα μετέπειτα μεταφοράς του σε ASIC.

Το έργο της παρούσας διπλωματικής εργασίας έγινε δεκτό για παρουσίαση στο κοινό UFFC, EFTF and PFM IEEE Symposium που θα πραγματοποιηθεί στη Πράγα στις 21-25 Ιουλίου 2013. Κατά τη διάρκεια του συμποσίου θα παρουσιαστεί η κατασκευή σε μορφή αφίσας, αλλά και θα γίνεται ζωντανή παρουσίαση του πομπού [1].

Abstract

In the context of this work, an FPGA circuit was constructed, implementing an all-digital video television transmitter, encoding Phase Alternating Line analog protocol. The circuit description was made using Verilog hardware description language and a Xilinx entry level FPGA board was used.

The technical features of the circuit include: Video capturing algorithms using a VGA 640x480 pixel, 30 fps camera, RAM based frame buffer, PAL encoding algorithms, amplitude modulation and multi to single-bit quantization using random number dithering techniques.

During the design process, main concern, apart from the flawless design, was a design that used the minimum resources on the FPGA board and could easily transposed onto an ASIC circuit.

The current work has been accepted and will be presented in the joint UFFC, EFTF and PFM IEEE Symposium which will be held in Prague on 21-25 of July 2013. A poster and a live demo will be presented, and a paper explaining this thesis work have been submitted, for this purpose [1].

Keywords

FPGA, Xilinx, Video Transmitter, PAL, Phase Alternating Line, All Digital, Verilog, Hardware Description Language, VLSI, Synthesis, place & route.

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε κατά το ακαδημαϊκό έτος 2012-2013 υπό την επίβλεψη του κ. Παύλου - Πέτρου Σωτηριάδη, επ. Καθηγητή του Ε.Μ.Π. της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, στον οποίο οφείλω ιδιαίτερες ευχαριστίες για την ανάθεσή της. Μέσα από τη διαδικασία υλοποίησης, αλλά και συγγραφής της παρούσας διπλωματικής εργασίας, είχα την ευκαιρία να εμβαθύνω σε διάφορους ενδιαφέροντες τομείς και να γίνω γνώστης του αντικειμένου της συγγραφής κώδικα περιγραφής υλικού καθώς και της τεχνολογίας των FPGA. Ιδιαίτερες ευχαριστίες θα ήθελα να απευθύνω στον υποψήφιο διδάκτορα κ. Κώστα Γαλανόπουλο που μου παρείχε την απαραίτητη βοήθεια και χωρίς αυτόν, αυτή η διπλωματική εργασία δεν θα ήταν δυνατό να ολοκληρωθεί. Τέλος θα ήθελα να ευχαριστήσω τους γονείς, την αδελφή και τον αδελφό μου και τους φίλους μου για τη στήριξη και βοήθεια καθ' όλη τη διάρκεια των σπουδών μου.

Περιεχόμενα

Κίνητρο	- 5 -
Περίληψη	- 6 -
Abstract	- 7 -
Τεχνικές Αναλογικής Διαμόρφωσης	- 12 -
1.1 Εισαγωγή:	- 12 -
1.2 Το μοντέλο Shannon:	- 13 -
1.2 Ζώνες Συχνοτήτων κατά ITU.....	- 14 -
1.3 Τεχνικές Αναλογικής Διαμόρφωσης:.....	- 15 -
1.3.1 Διαμόρφωση Πλάτους Διπλής Πλευρικής Ζώνης (AM-DSB):.....	- 15 -
1.3.2 Διαμόρφωση Πλάτους Απλής Πλευρικής Ζώνης (SSB).....	- 18 -
1.3.3 Διαμόρφωση Υπολειπόμενης Πλευρικής Ζώνης (VSB).....	- 19 -
Συστήματα Αναλογικής Κωδικοποίησης Τηλεοπτικού Σήματος	- 21 -
2.1 Ιστορική Αναδρομή	- 21 -
2.2 Σύστημα NTSC (National Television System Committee)	- 22 -
2.3 Σύστημα SECAM (Séquentiel couleur à mémoire).....	- 23 -
2.4 Σύστημα PAL (Phase Alternating Line).....	- 24 -
2.5 Αναλυτική Περιγραφή Συστήματος PAL:.....	- 25 -
2.6 Το πρωτόκολλο VGA	- 33 -
2.7 Η κάμερα OV7670.....	- 33 -
Εισαγωγή στη τεχνολογία FPGA	- 35 -
3.1 Εισαγωγή	- 35 -
3.2 Συστοιχίες επιτόπια προγραμματιζόμενων πυλών (FPGAs)	- 35 -
3.3 Ιστορική Αναδρομή	- 36 -
3.4 Αρχιτεκτονική FPGA.....	- 36 -
3.5 Διαδικασία Σχεδίασης/ Υλοποίησης προγράμματος σε FPGA	- 38 -
3.4 Οικογένεια FPGA Xilinx® Spartan™ -3E.....	- 39 -
3.4.1 Εισαγωγή	- 39 -
3.4.2 Αρχιτεκτονική Xilinx® Spartan™ -3E	- 39 -
Υλοποίηση του πομπού σε τεχνολογία FPGA	- 48 -
4.1 Περισκόπηση συστήματος	- 48 -
4.2 Υλοποίηση συστήματος σε υλικό	- 48 -
4.2.1 Υπομονάδα PAL (PAL Module).....	- 49 -
4.2.1.1 Υπομονάδα Line_Gen	- 50 -
4.2.1.2 Μηχανή Πεπερασμένων Καταστάσεων PAL.....	- 50 -
4.2.1.3 Υπομονάδα VGAtοPAL (VGAtοPAL Module).....	- 54 -

4.2.2.3.α Υπομονάδα CAMtoRAM	- 55 -
4.2.2.3.β Υπομονάδα RAM	- 56 -
4.2.2.3.γ Υπομονάδα RAMtoPAL	- 57 -
4.2.3 Διαμόρφωση Πλάτους (AM).....	- 59 -
4.2.3.α SineGen module	- 60 -
4.2.4 Μετατροπή ψηφιακού σήματος σε αναλογικό (Random number Dithering)	- 61 -
4.2.4.α Γεννήτρια τυχαίων αριθμών (RAND module).....	- 62 -
Μετρήσεις, αποτελέσματα και μελλοντικές επεκτάσεις	- 63 -
5.1 Σύνοψη:.....	- 63 -
5.2 Μελλοντικές Επεκτάσεις	- 65 -
Παράρτημα – Κώδικας υλοποίησης σε Verilog	- 67 -
Βιβλιογραφία	- 79 -

1

Τεχνικές Αναλογικής Διαμόρφωσης

1.1 Εισαγωγή:

“Επικοινωνία είναι όλες εκείνες οι διαδικασίες που εμπλέκονται στη μεταφορά της πληροφορίας από τον αποστολέα στον παραλήπτη”

Claude Elwood Shannon

Η επικοινωνία είναι ένα σύνθετο φαινόμενο άρρηκτα συνδεδεμένο με το ανθρώπινο είδος. Η ιστορία των τηλεπικοινωνιών χωρίζεται σε δύο μεγάλες περιόδους: πριν και μετά την εμφάνιση του ηλεκτρισμού. Πριν την εμφάνιση του ηλεκτρισμού οι τηλεπικοινωνίες στην αρχαία Ελλάδα γίνονταν με μέσα όπως: [1]

- Φωτεινοί Αναμεταδότες
- Οπτικός Τηλέγραφος
- Ακουστικό κέρας Μ.Αλεξάνδρου
- Υδραυλικός Τηλέγραφος
- Ταμ-Ταμ
- Σήματα Καπνού
- Ταχυδρομικά Περιστερία
- Μηχανικός Τηλέγραφος

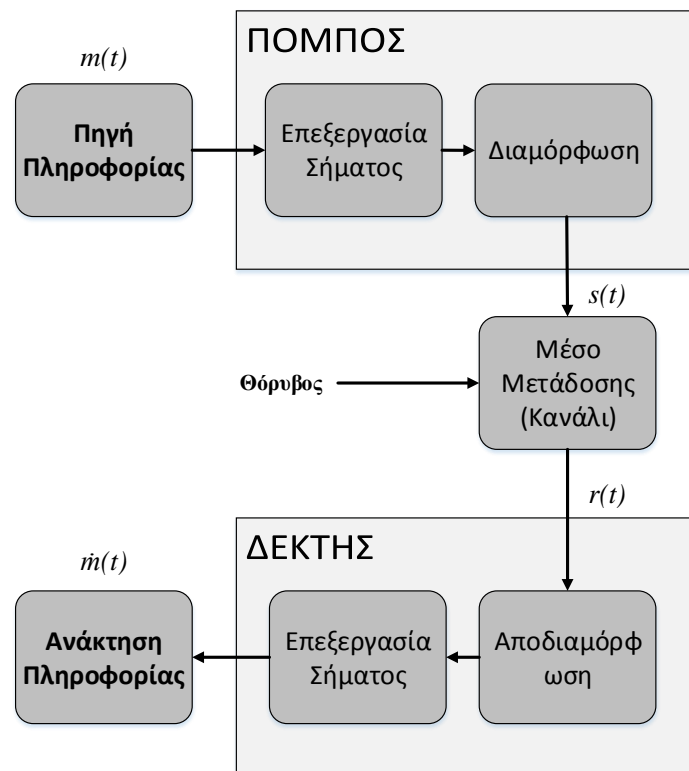
Η εμφάνιση του ηλεκτρισμού έφερε επανάσταση στις τηλεπικοινωνίες. Η πρώτη εφεύρεση, ο τηλέγραφος έκανε της εμφάνισή της το 1839. Έκτοτε οι εφευρέσεις στον τομέα των τηλεπικοινωνιών έκαναν τεράστια άλματα. Παρακάτω παρουσιάζονται οι σημαντικότερες στιγμές στον τομέα των τηλεπικοινωνιών μετά την ανακάλυψη του ηλεκτρισμού:

1837	Cooke και Wheatstone αποκτούν ευρεσιτεχνία για τον τηλέγραφο. Ο Samuel F.B Morse επιδεικνύει δημόσια το δικό του τηλέγραφο.
1866	Πρώτη υπερατλαντική επικοινωνία μέσω δύο υποβρυχίων καλωδίων τηλέγραφου.
1884	Ο Paul Gottlieb Nipkow αποκτά ευρεσιτεχνία για την κατασκευή της πρώτης ηλεκτρομηχανικής τηλεόρασης.
1876	Οι Alexander Graham Bell και Elisha Gray εφευρίσκουν ανεξάρτητα τον τηλέγραφο. Ο Bell κερδίζει δικαστικά την προτεραιότητα.
1887	Heinrich Hertz: Επαλήθευση της θεωρίας του Maxwell περί της ύπαρξης και διάδοσης ηλεκτρομαγνητικών κυμάτων.
1893	Nikolai Tesla: Παρουσίασε και επέδειξε τις βασικές αρχές ενός ασύρματου τηλεπικοινωνιακού συστήματος

1895	Ο Guglielmo Marconi στέλνει τα πρώτα υπερατλαντικά ασύρματα σήματα
1906	Lee de Forest: Τρίοδος ενισχυτική λυχνία κενού. Κατέστησε δυνατή την εισαγωγή ενισχυτών σήματος στα τηλεφωνικά συστήματα και συνεπώς επέτρεψε την μετάδοση τους σε μεγαλύτερες αποστάσεις. Έπαιξε σημαντικό ρόλο στην ανάπτυξη της ραδιοφωνίας και των τηλεπικοινωνιών εν γένει.
1920	Πραγματοποιείται η πρώτη ραδιοφωνική μετάδοση AM
1928	Ο Philo Farnsworth κατασκευάζει την πρώτη τηλεόραση καθοδικού σωλήνα.
1953	Ο John Pierce προτείνει τις επικοινωνίες βαθέως διαστήματος (deep space communications)
1957	Η Σοβιετική Ένωση εκτοξεύει τον Sputnik, τον πρώτο τεχνητό δορυφόρο.
1962	Τέθηκε σε τροχιά από τη NASA ο πρώτος πειραματικός δορυφόρος Telstart I που μετέδιδε τηλεοπτικό σήμα
1979	Πρώτη εμπορικά διαθέσιμη τεχνολογία αναλογικής (κινητής) κυψελωτής τηλεφωνίας 1G
1991	Ψηφιακή κυψελωτή κινητή τηλεφωνία δεύτερης γενιάς (2G) στη Φινλανδία και αλλού. Μέχρι σήμερα είναι η επικρατέστερη τεχνολογία κινητής τηλεφωνίας.
2001	Ψηφιακή κυψελωτή κινητή τηλεφωνία υψηλού ρυθμού μετάδοσης δεδομένων και πολυμέσων (3G) στο Τόκιο

1.2 Το μοντέλο Shannon:

Το πιο γνωστό επικοινωνιακό μοντέλο είναι αυτό που παρουσίασαν οι C.E.Shannon και W.Weaver στα εργαστήρια της Bell το 1949. Το μοντέλο το οποίο χαρακτηρίζεται «μητέρα όλων των μοντέλων» εισάγει τις έννοιες της πηγής πληροφορίας (information source), του μηνύματος (information message), του πομπού (transmitter), του δέκτη (receiver) και του καναλιού. Παρακάτω παρουσιάζεται το διάγραμμα ενός τηλεπικοινωνιακού συστήματος.



Σχήμα 1.1: Σχεδιάγραμμα τηλεπικοινωνιακού συστήματος

Ανεξάρτητα από την εφαρμογή, κάθε τηλεπικοινωνιακό σύστημα αποτελείται από τις τρεις βασικές μονάδες: τον Πομπό, το Κανάλι και το Δέκτη.

- **Πομπός:** Ο πομπός αποτελείται από τις βαθμίδες Επεξεργασίας Σήματος και Διαμόρφωσης. Το σήμα πληροφορίας που εισέρχεται στον πομπό $m(t)$ μπορεί να είναι αναλογικό ή ψηφιακό και μπορεί να είναι σήμα φωνής, εικόνας, κειμένου κ.τ.λ.

Το φάσμα του $m(t)$ είναι συγκεντρωμένο γύρω από τη μηδενική συχνότητα και καλείται σήμα βασικής ζώνης (baseband).

Κατά την επεξεργασία σήματος, το σήμα κωδικοποιείται με τέτοιο τρόπο ώστε να αντιμετωπίσει με τον καλύτερο δυνατό τρόπο τις αλλοιώσεις που υφίσταται από το δίαυλο επικοινωνίας (κανάλι).

Η βαθμίδα διαμόρφωσης (modulation) μετατρέπει το σήμα βασικής ζώνης (baseband), σε ζωνοπερατό (passband), δηλαδή σε συχνότητα $f_c \gg 0$ έτσι ώστε να μεταδοθεί μέσω του καναλιού.

- **Κανάλι:** Το κανάλι είναι το μέσο μετάδοσης της πληροφορίας από τον πομπό στο δέκτη. Κατά την μετάδοση, το κανάλι προκαλεί εξασθένιση και εισάγει θόρυβο, προκαλώντας αλλοίωση στο εκπεμπόμενο σήμα. Ο θόρυβος μπορεί να προκαλείται από διάφορες αιτίες και προκαλεί διαφορετικές αλλοιώσεις στο εκπεμπόμενο σήμα.

Παραδείγματα θορύβου είναι ο θερμικός θόρυβος, ο θόρυβος που οφείλεται στην ανθρώπινη παρουσία, ο θόρυβος από το διάστημα (κοσμική ακτινοβολία) κ.λ.π.

Το κανάλι μπορεί να είναι μια γραμμή μεταφοράς (τηλεφωνία, τηλεγραφία), οπτική ίνα (οπτικές επικοινωνίες ή απλά ο ελεύθερος χώρος όπου το σήμα εκπέμπεται σαν ηλεκτρομαγνητικό κύμα (ραδιοφωνία, τηλεόραση, δορυφορικές επικοινωνίες).

- **Δέκτης:** Σκοπός του δέκτη είναι η αναδημιουργία του αρχικού σήματος πληροφορίας από το υποβαθμισμένο σήμα που φτάνει σε αυτόν. Αυτή η αναδημιουργία επιτυγχάνεται μέσω της διαδικασίας της αποδιαμόρφωσης που είναι η αντίστροφη από αυτή της διαμόρφωσης. Το λαμβανόμενο σήμα μέσω αυτής της διαδικασίας μεταβαίνει από την ζωνοπερατή στη βασική συχνότητα.

Παρόλα αυτά, λόγω της παρουσίας θορύβου, ο δέκτης δεν μπορεί να αναδημιουργήσει το αρχικό σήμα επακριβώς.

Η συνολική υποβάθμιση που υπόκειται το σήμα πληροφορίας, επηρεάζεται από τον τύπο διαμόρφωσης, καθώς μερικά σήματα διαμόρφωσης είναι λιγότερο ευαίσθητα στις επιδράσεις του θορύβου και τη της παραμόρφωσης.

1.2 Ζώνες Συχνότητων κατά ITU

Η Διεθνής Ένωση Τηλεπικοινωνιών (International Telecommunication Union, ITU), ο αρμόδιος οργανισμός για τις τηλεπικοινωνίες, έχει καθορίσει τις ζώνες ραδιοσυχνότητων και τη χρήση υπηρεσιών που θα γίνεται σε κάθε ζώνη [3]. Στον Πίνακα 1.1 παρουσιάζονται οι ζώνες συχνότητων και οι συνήθεις χρήσεις τους. Όπως βλέπουμε η υπηρεσίες τηλεόρασης καταλαμβάνουν συχνότητες στις μπάντες των VHF (Very High Frequencies) και UHF (Ultra High Frequencies).

Ζώνη Συχνότητας	Ονομασία	Υπηρεσίες
3-30 KHz	VLF (Very Low Frequencies)	Συστήματα Πλοήγησης Sonar
30-300 KHz	LF (Low Frequencies)	Ραδιοφάροι Βοηθήματα Πλοήγησης
300-3000 KHz	MF (Medium Frequencies)	Ραδιοφωνία AM, Ραδιοεπικοινωνίες Ναυσιπλοΐας
3-30 MHz	HF (High Frequencies)	Τηλεφωνία, Τηλεγραφία, Τηλεομοιοτυπία, Ερασιτεχνική Ραδιοφωνία, Επικοινωνίες μεταξύ πλοίων - αεροπλάνων και σταθμών εδάφους
30-300 MHz	VHF (Very High Frequencies)	Τηλεόραση, Ραδιοφωνία FM, Συστήματα Εναέριας Κυκλοφορίας, CBs (Citizens' Band radio)
300-3000 MHz	UHF (Ultra High Frequencies)	Τηλεόραση, Δορυφορικές Επικοινωνίες, Συστήματα Radar, Κινητές Επικοινωνίες
3-30 GHz	SHF (Super High Frequencies)	Συστήματα Radar, Μικροκομματικές Επικοινωνίες, Δορυφορικές Ζεύξεις, Κινητές Επικοινωνίες, Wireless LAN, Bluetooth
3-300 GHz	EHF (Extremely High Frequencies)	Συστήματα Radar
300-3000 GHz	THz (Terahertz) ή THF (Tremendously High Frequency)	Ιατρικές εφαρμογές, Τηλεπικοινωνίες πολύ ψηλού υψομέτρου

1.3 Τεχνικές Αναλογικής Διαμόρφωσης:

Η μετάδοση αναλογικού σήματος τηλεόρασης γίνεται με τεχνικές αναλογικής διαμόρφωσης σήματος και συγκεκριμένα με συνδιασμό των μεθόδων διαμόρφωσης πλάτους, ορθογωνικής διαμόρφωσης πλάτους και διαμόρφωσης συχνότητας. Στο παρόν κεφάλαιο θα αναλύσουμε τις τεχνικές αναλογικής διαμόρφωσης σήματος και της συνήθεις μεθόδους υλοποίησής τους σε υλικό.

1.3.1 Διαμόρφωση Πλάτους Διπλής Πλευρικής Ζώνης (AM-DSB):

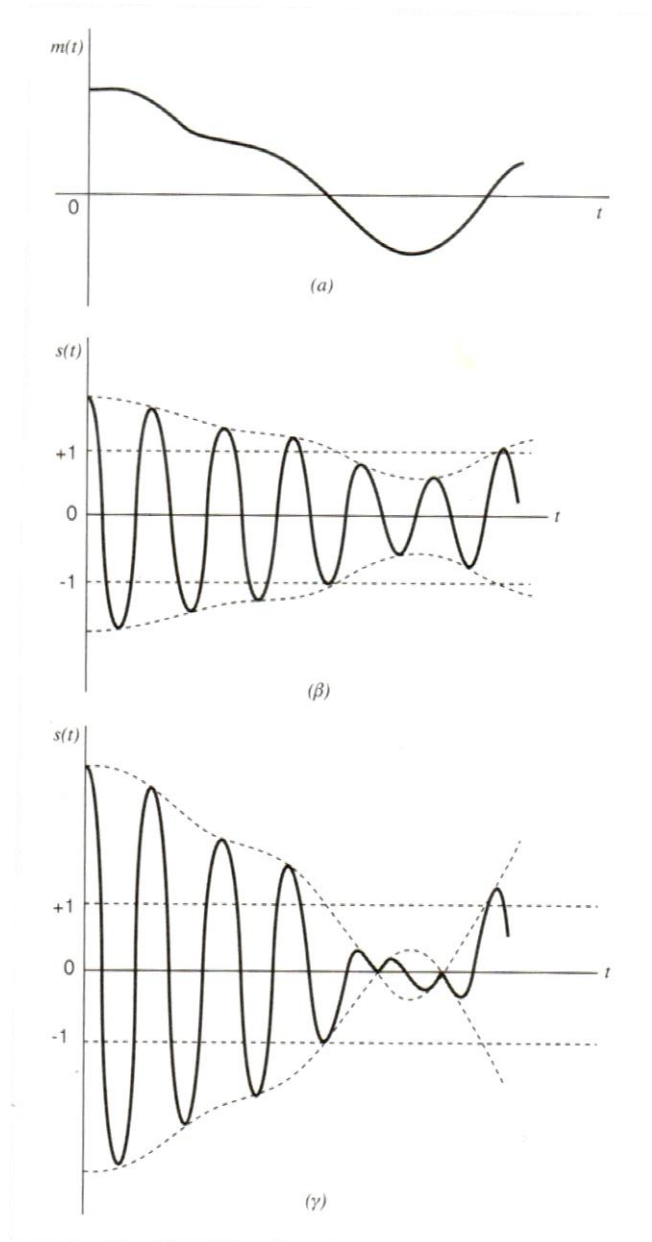
Θεωρούμε ημιτονικό φέρον της μορφής:

$$c(t) = A_c \cos(2\pi f_c t)$$

όπου A_c το πλάτος και f_c η συχνότητα του φέροντος και θεωρείται η φάση του φέροντος ίση με μηδέν. Αν $m(t)$ είναι το σήμα βασικής ζώνης και θεωρείται ανεξάρτητο από το φέρον $c(t)$, η διαμόρφωση πλάτους (Amplitude Modulation, AM) ορίζεται σαν η διαδικασία στην οποία το πλάτος του φέροντος $c(t)$ μεταβάλλεται γύρω από μια μέση τιμή γραμμικά σε σχέση με το σήμα βασικής ζώνης $m(t)$. Μια κατά πλάτος διαμορφωμένη κυματομορφή, στο πεδίο του χρόνου, δίνεται από την παρακάτω εξίσωση:

$$s(t) = A_c [1 + k_a m(t)] \cos(2\pi f_c t)$$

όπου k_a είναι σταθερά που καλείται ευαισθησία πλάτους (amplitude sensitivity) του διαμορφωτή.



Σχήμα 1.2: Διαδικασία διαμόρφωσης πλάτους.

(α) Σήμα βασικής ζώνης $m(t)$,

(β) Διαμορφωμένη κυματομορφή με $|k_a m(t)| < 1$ για κάθε t .

(γ) Διαμορφωμένη κυματομορφή με $|k_a m(t)| > 1$ για μερικά t .

Στο Σχήμα 1.2 βλέπουμε τη διαμόρφωση που υπόκειται το σήμα βασικής ζώνης $m(t)$ (α) και τις αντίστοιχες AM διαμορφωμένες κυματομορφές για δύο τιμές ευαισθησίας πλάτους (β) και (γ). Για τιμές του $|k_a m(t)| < 1$ (β) παρατηρούμε ότι η περιβάλλουσα $s(t)$ έχει την ίδια μορφή με το σήμα βασικής ζώνης. Για τιμές του $|k_a m(t)| > 1$ για κάποια t (γ) η περιβάλλουσα $s(t)$ δεν έχει την ίδια μορφή και παρουσιάζεται αντιστροφή φάσης κάθε φορά που ο παράγοντας $1+k_a m(t)$ αλλάζει πρόσημο. Τότε λέμε πως η φέρουσα κυματομορφή υπόκειται σε υπερδιαμόρφωση (over-modulation) [4].

Γενικά για να έχουμε διαμόρφωση πλάτους χωρίς παραμορφώσεις πρέπει να ισχύουν τα εξής:

α. $|k_a m(t)| \leq 1$

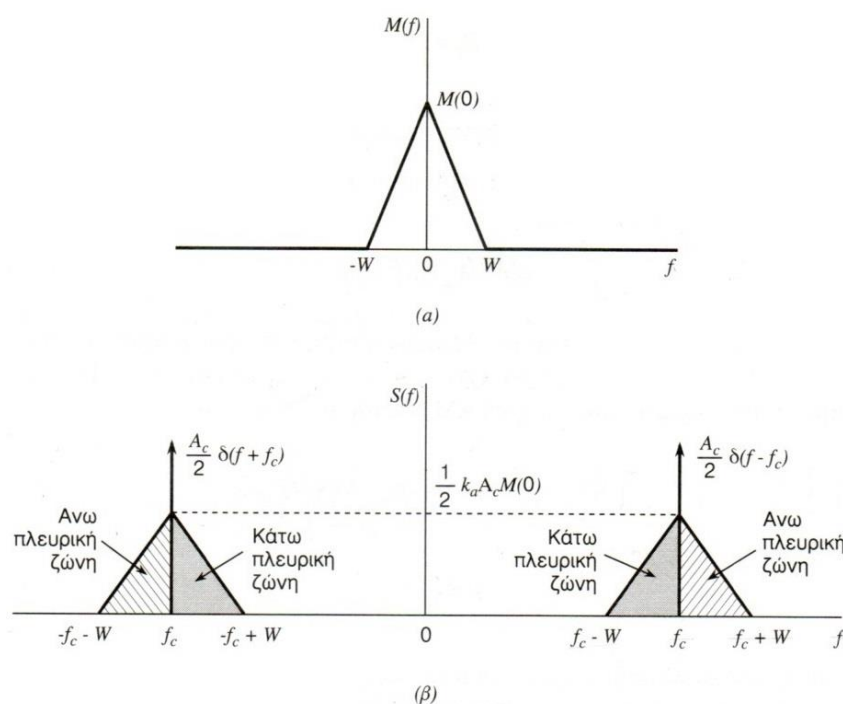
β. $f_c \gg W$

όπου W το εύρος ζώνης πληροφορίας.

Ο μετασχηματισμός Fourier της AM κυματομορφής $s(t)$ δίνεται από τον τύπο:

$$S(f) = \frac{A_c}{2} [\delta(f - f_c) + \delta(f + f_c)] + \frac{k_a A_c}{2} [M(f - f_c) + M(f + f_c)]$$

Αν το σήμα βασικής ζώνης έχει εύρος ζώνης από $-W$ έως $+W$ και $f_c > W$, το φάσμα της $S(f)$ παρουσιάζεται στο Σχήμα 1.3 (β). Το σήμα αποτελείται από δύο συναρτήσεις δέλτα με συντελεστή βαρύτητας $A_c/2$ που εμφανίζονται στις συχνότητες $\pm f_c$ και δύο εκδοχές του φάσματος με πλάτος $k_a A_c/2$ γύρω από τις συχνότητες $\pm f_c$.



Σχήμα 1.3: (α) Φάσμα σήματος Βασικής Ζώνης
(β) Φάσμα κυματομορφής AM

Το μέρος του φάσματος που βρίσκεται πάνω από τη συχνότητα φέροντος καλείται άνω πλευρική ζώνη (upper sideband), ενώ το μέρος κάτω από τη συχνότητα φέροντος καλείται κάτω πλευρική ζώνη (lower sideband). Για να εξασφαλίζεται ότι οι πλευρικές ζώνες δεν αλληλοκαλύπτονται πρέπει να ισχύει η ακόλουθη σχέση: $f_c > W$

Η χαμηλότερη συνιστώσα που μεταδίδεται είναι ίση με $f_c - W$, ενώ η υψηλότερη είναι ίση με $f_c + W$. Επομένως το εύρος ζώνης μετάδοσης (transmission bandwidth) είναι:

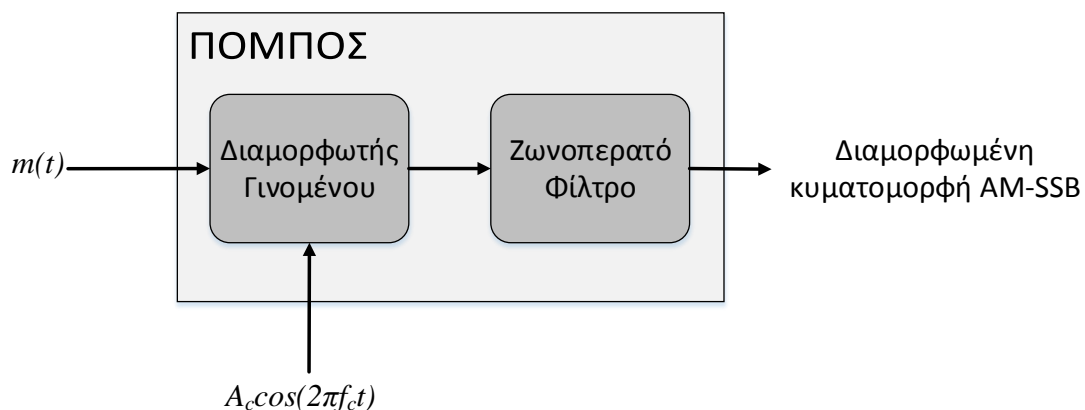
$$B_T = 2W$$

1.3.2 Διαμόρφωση Πλάτους Απλής Πλευρικής Ζώνης (SSB)

Όπως είδαμε, οι διαμορφώσεις Διπλής Πλευρικής Ζώνης (Double Sideband, DSB), καταλαμβάνουν διπλάσιο εύρος ζώνης, από το σήμα βασικής ζώνης, καταλαμβάνοντας πολύτιμο χώρο στο φάσμα ραδιοσυχνοτήτων, ενώ αρκεί η μετάδοση μόνο μιας από τις δύο πλευρικές ζώνες για ανάκτηση της πληροφορίας από τη δέκτη.

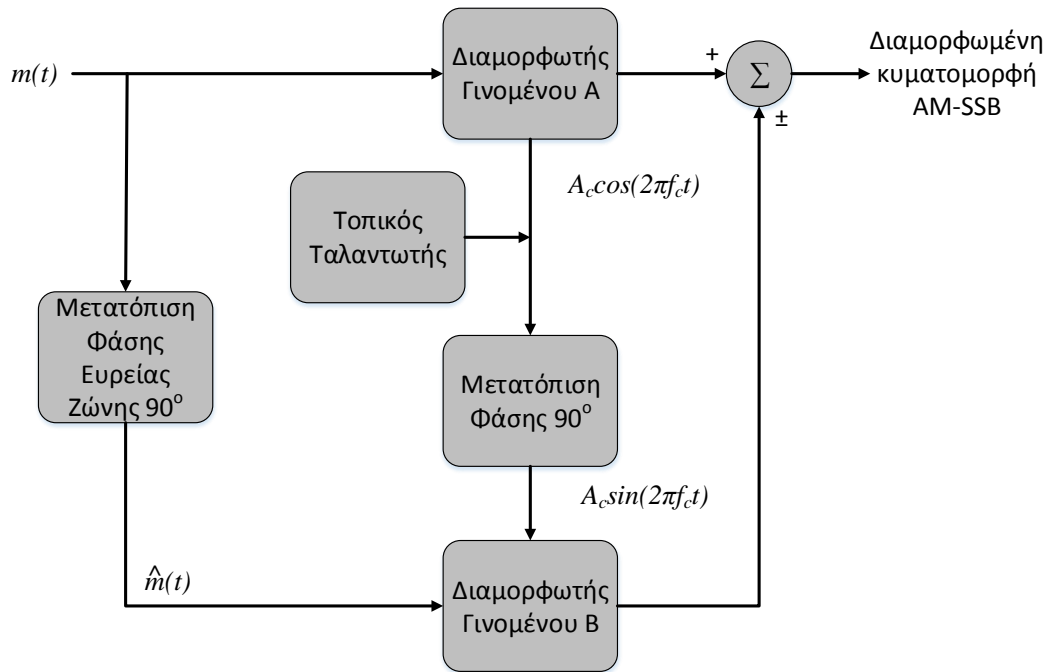
Αυτή η διαπίστωση οδήγησε στη Διαμόρφωση Απλής Πλευρικής Ζώνης (Single Sideband Amplitude Modulation, SSB-AM), στην οποία χρησιμοποιείται μόνο η μία από τις δύο πλευρικές ζώνες [5].

Υπάρχουν δύο συνήθεις τρόποι για επίτευξη μετάδοσης διαμόρφωσης πλάτους απλής πλευρικής ζώνης. Ο πρώτος τρόπος, Σχήμα 1.4, είναι με τη μέθοδο διεκρίνησης συχνότητας (frequency discrimination method), με χρήση κατάλληλου ζωνοπερατού φίλτρου, που επιτρέπει μόνο στην μια πλευρική ζώνη να περάσει, απορρίπτοντας την άλλη.



Σχήμα 1.4: Μέθοδος διεκρίνησης συχνότητας

Ο δεύτερος τρόπος είναι με τη χρήση δύο διαμορφωτών γινομένου και στη συνέχεια με τον συνδυασμό των δύο προϊόντων διαμόρφωσης που προκύπτουν, Σχ.1.5. Οι δύο διαμορφωτές τροφοδοτούνται από φέροντα που έχουν διαφορά φάσης 90° μεταξύ τους. Το εισερχόμενο σήμα πληροφορίας βασικής ζώνης $m(t)$ εφαρμόζεται στο διαμορφωτή γινομένου A παράγοντας μια κυματομορφή DSCB-SC που έχει πλευρικές ζώνες φάσης αναφοράς (reference phase), συμμετρικά τοποθετημένες γύρω από το φέρον f_c . Ο μετασχηματισμός Hilbert $\hat{m}(t)$ του $m(t)$ εφαρμόζεται στο διαμορφωτή γινομένου B, δημιουργώντας μια κυματομορφή DSB-SC που περιέχει πλευρικές ζώνες με πλάτη όμοια με εκείνα του διαμορφωτή γινομένου A, αλλά με φάση τέτοια έτσι ώστε η διανυσματική πρόσθεση ή αφαίρεση των δύο εξόδων, των διαμορφωτών στον αθροιστή να έχει σαν αποτέλεσμα την αναίρεση της μιας πλευρικής ζώνης και την ενίσχυση της άλλης. Η άθροιση της εξόδου του διαμορφωτή γινομένου B με την έξοδο του διαμορφωτή γινομένου A, δίνει κυματομορφή SSB με μόνο την κάτω πλευρική ζώνη, ενώ η αφαίρεση δίνει μόνο την άνω πλευρική ζώνη. Ο διαμορφωτής αυτός είναι γνωστός και σαν διαμορφωτής Hartley.



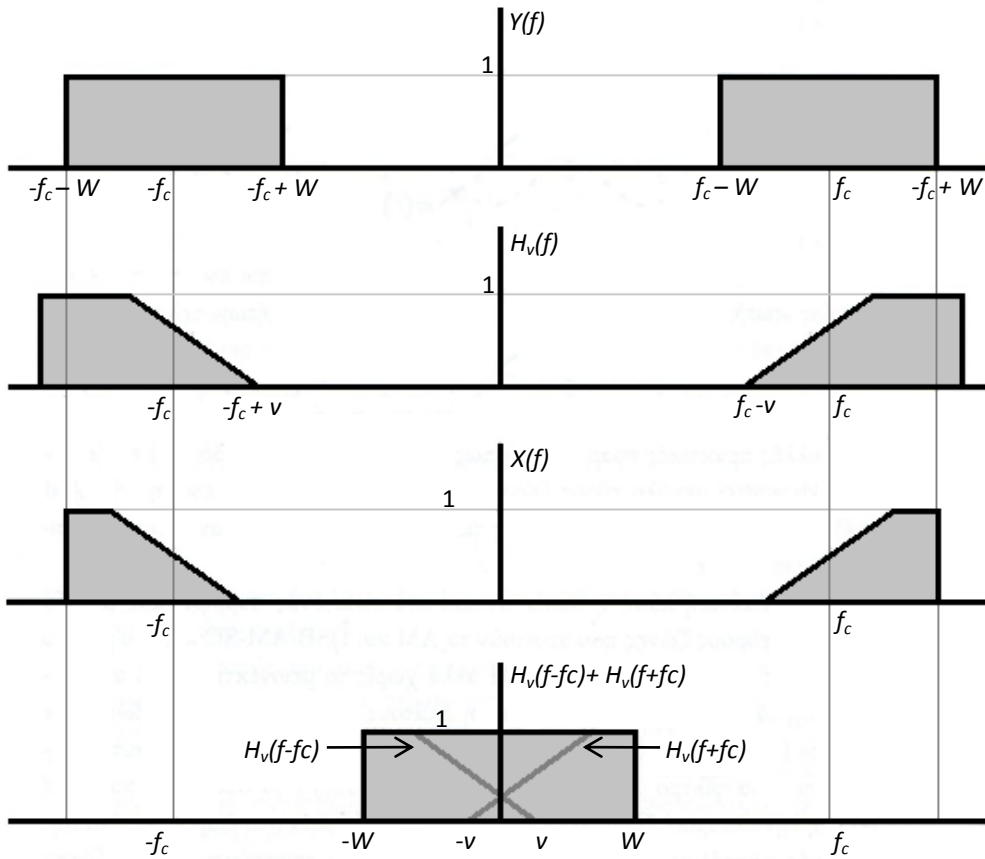
Σχήμα 1.5: Μέθοδος Διευκρίνησης Φάσματος (Διαμορφωτής Hartley)

1.3.3 Διαμόρφωση Υπολειπόμενης Πλευρικής Ζώνης (VSB)

Η διαμόρφωση απλής πλευρικής ζώνης (SSB) είναι αρκετά βολική για μετάδοση ήχου, αφού οι ακούστικες συχνότητες έχουν κενό μεταξύ του μηδέν και μερικών εκατοντάδων Hertz. Στη περίπτωση της τηλεόρασης, που το σήμα βασικής ζώνης περιέχει συνιστώσες σε χαμηλές συχνότητες και οι άνω και κάτω πλευρικές συνιστώσες συναντιούνται στη συχνότητα φέροντος, η διαμόρφωση AM-SSB δεν ενδείκνυται, γιατί θα απέκοπτε μέρος από τη χρήσιμη πληροφορία. Η διαμόρφωση AM-DSB χρησιμοποιεί διπλάσιο εύρος ζώνης σε σχέση με την AM-SSB, οπότε ούτε αυτός ο τύπος διαμόρφωσης ενδείκνυται. Μια εναλλακτική λύση είναι ο συμβιβασμός μεταξύ AM-SSB και AM-DSB, η διαμόρφωση υπολειπόμενης πλευρικής ζώνης (Vestigial Sideband Modulation, VSB). Σε αυτό το σύστημα η μια πλευρική ζώνη περνάει ολόκληρη, ενώ από την άλλη πλευρική ζώνη περνάει ένα κατάλοιπο (vestige). Έτσι το εύρος ζώνης που απαιτείται είναι μικρότερο από αυτό της διαμόρφωσης AM-DSB αλλά ταυτόχρονα λίγο μεγαλύτερο από αυτό της AM-SSB. Έτσι γίνεται μετάδοση σημάτων με φασματικό περιεχόμενο που ξεκινά από το μηδέν, χωρίς παραμόρφωση και με το χρήση του ελάχιστου δυνατού εύρους ζώνης.

Η διαμόρφωση VSB γίνεται με την επιλογή του κατάλληλου ζωνοπερατού φίλτρου $H_u(f)$ όπως φαίνεται στο Σχ. 1.6. Το φάσμα $S(f)$ της κυματομορφής $s(t)$ δίνεται από τον παρακάτω τύπο:

$$S(f) = \frac{A_c}{2} [M(f - f_c) + M(f + f_c)] H(f)$$

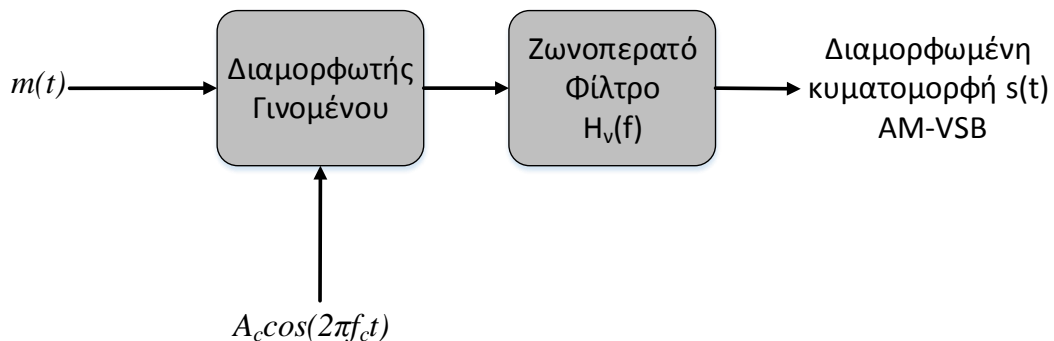


Σχήμα 1.6: Διαμόρφωση Υπολειπόμενης Πλευρικής Ζώνης (AM-VSB)
 (α) Φάσμα διαμορφωμένου DSB σήματος
 (β) Συνάρτηση μεταφοράς φίλτρου VSB
 (γ) Εκπεμπόμενο Φάσμα
 (δ) Αποδιαμόρφωση VSB

Το εύρος ζώνης μετάδοσης που απαιτείται από το σύστημα VSB είναι:

$$B_T = W + f_v$$

Όπου W είναι το εύρος ζώνης της ολόκληρης πλευρικής ζώνης και f_v το εύρος της υπολειπόμενης πλευρικής ζώνης.



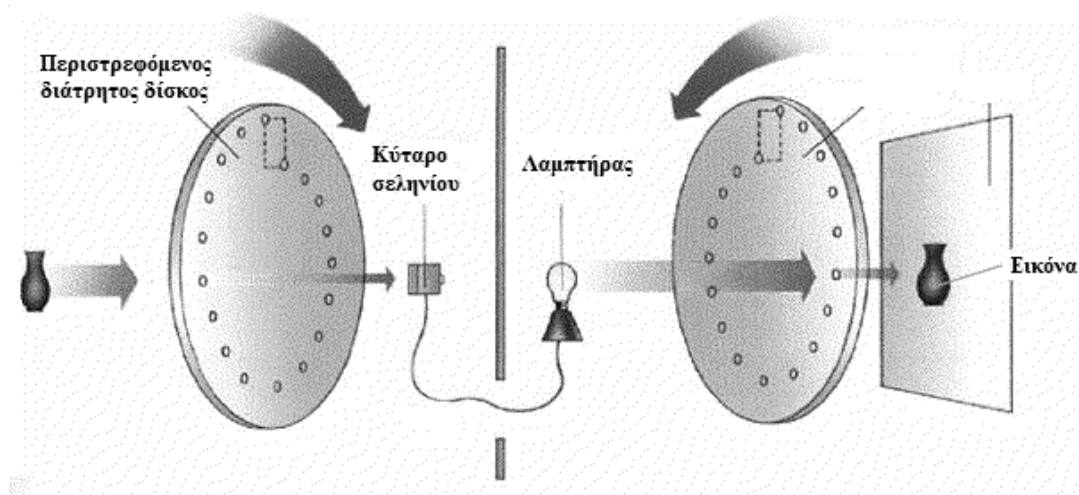
Σχήμα 1.7: Δομικό διάγραμμα μεθόδου Διαμόρφωσης Υπολειπόμενης Πλευρικής Ζώνης

2

Συστήματα Αναλογικής Κωδικοποίησης Τηλεοπτικού Σήματος

2.1 Ιστορική Αναδρομή

Η αναλογική τηλεόραση είναι έργο πολλών μηχανικών και επιστημόνων σε διάφορες χώρες. Πρώτος ο A. A. Campbell-Swinton πρότεινε την αρχή λειτουργίας της τηλεόρασης σε δημοσίευσή του το 1908, κάνοντας χρήση οθονών σωλήνα καθοδικών ακτινών. Πριν την ανακάλυψη των σωλήνων καθοδικών ακτινών, μια ηλεκτρομηχανική εκδοχή της τηλεόρασης είχε παρουσιαστεί από τον 23χρονο φοιτητή Paul Nipkow το 1898 [2]. Η συσκευή αποτελούνταν από ένα περιστρεφόμενο δίσκο με τρύπες σε ίση απόσταση μεταξύ τους, τοποθετημένες σε ελικοειδή διάταξη. Καθώς ο δίσκος περιστρεφόταν με σταθερή ταχύτητα, οι τρύπες επέτρεπαν το φως από διαφορετικό σημείο του εικονιζόμενου αντικειμένου να προσκρούσει σε ένα αισθητήρα φωτός (κύταρο σεληνίου). Έτσι η εικόνα κωδικοποιούνταν σε αναλογικό ηλεκτρικό σήμα που μεταδιδόταν ενσύρματα, ή ασύρματα στο δέκτη. Ο δέκτης αποτελούνταν και αυτός από ένα περιστρεφόμενο δίσκο, ένα λαμπτήρα και την οθόνη. Η ένταση του λαμπτήρα αυξομειωνόταν με το μεταδιδόμενο σήμα, και περνώντας από τον περιστρεφόμενο δίσκο σχημάτιζε την αρχική εικόνα στην οθόνη.

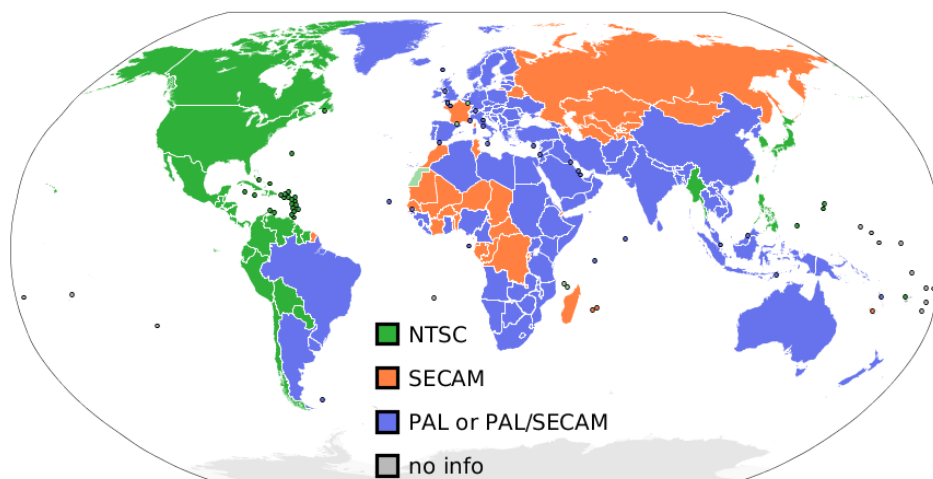


Σχήμα 2.1 : Ηλεκτρομηχανική Συσκευή Τηλεόρασης

Αργότερα η ανακάλυψη του σωλήνα καθοδικών ακτινών έδωσε την ευκαιρία η τηλεόραση να μετατραπεί από μια ηλεκτρομηχανική σε μια καθαρά ηλεκτρική συσκευή. Η πρώτη μετάδοση εικόνας με τη χρήση αυτής της συσκευής έγινε το 1925 από το Σκωτσέζο εφευρέτη John Logie Baird στο εργαστήριό του. Τον ίδιο χρόνο ο Charles Francis Jenkins επιτυγχάνει την

πρώτη ασύρματη μετάδοση εικόνας, ενώ το 1929 εκπέμπει ο πρώτος πειραματικός τηλεοπτικός σταθμός, σε συνεργασία με το BBC.

Μέχρι το 1951 η εκπομπή τηλεοπτικού σήματος γινόταν με ασπρόμαυρη εικόνα. Αργότερα, η ανακάλυψη της δυνατότητας εκπομπής χρώματος, καθιστά αναγκαία την προτυποποίηση των συστημάτων αναλογικής κωδικοποίησης εικόνας και ήχου. Σήμερα ανά τον κόσμο έχουν καθιερωθεί τρία συστήματα, με αρκετές διαφοροποιήσεις ανά χώρα. Στο παρόν κεφάλαιο επιχειρείται η παρουσίαση τριών προτύπων και ειδικά αναλυτική παρουσίαση του προτύπου PAL, που χρησιμοποιείται στην Ελλάδα και υλοποιήθηκε στην παρούσα διπλωματική εργασία.



Σχήμα 2.2 : Συστήματα αναλογικής κωδικοποίησης τηλεοπτικού σήματος ανά χώρα

2.2 Σύστημα NTSC (National Television System Committee)

Το σύστημα NTSC (National Television System Committee) αναπτύχθηκε το 1940 στις Ηνωμένες Πολιτείες από την FCC (Federal Communication Commission), με σκοπό να διαλύσει τις διαφορές μεταξύ των εταιριών κατασκευής τηλεοράσεων. Αρχικά το πρότυπο προοριζόταν για μετάδοση ασπρόμαυρης εικόνας, ενώ το 1953 επεκτάθηκε και για μετάδοση έγχρωμης εικόνας [7].

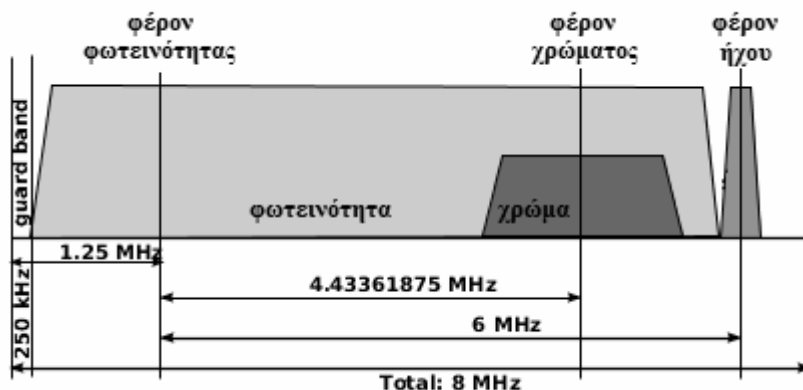
Χρησιμοποιείται κυρίως στη Βόρεια Αμερική (Καναδάς, Μεξικό, Ηνωμένες Πολιτείες Αμερικής), Νότιο Αμερική (εκτός των χωρών: Βραζιλία, Αργεντινή Ουρουγουάη και Γαλλική Γουινέα), Μιανμάρ, Νότιο Κορέα, Ταϊβάν, Ιαπωνία, Φιλιπίνες και περιοχές του Ειρηνικού Ωκεανού.

Το πρωτόκολλο NTSC αποτελείται από 29.97 καρέ ανά δευτερόλεπτο. Κάθε εικόνα χωρίζεται σε δύο πλαίσια των 262.5 γραμμών, 525 γραμμές στο σύνολο. Οι 483 γραμμές είναι στο ορατό μέρος της οθόνης, ενώ οι υπόλοιπες χρησιμοποιούνται για συγχρονισμό και για επιστροφή της δέσμης ηλεκτρονίων της τηλεόρασης στην πρώτη γραμμή. Ο ρυθμός ανανέωσης είναι περίπου 59.94 Hz (60 Hz / 1.001) για αποφυγή των φαινομένων ενδοδιαμόρφωσης μεταξύ των φέροντων της φωτεινότητας, του χρώματος και του ήχου. Σε σύγκριση με τα άλλα δύο συστήματα το NTSC έχει ψηλότερο ρυθμό ανανέωσης (30 καρέ ανά δευτερόλεπτο αντί 25) αλλά μικρότερη κάθετη ανάλυση.

2.3 Σύστημα SECAM (Séquentiel couleur à mémoire)

Το σύστημα SECAM (Séquentiel couleur à mémoire) που μεταφράζεται σε “ακολουθιακό χρώμα με μνήμη” υιοθετήθηκε το 1956 στη Γαλλία και θεωρείται το πρώτο Ευρωπαϊκό σύστημα κωδικοποίησης τηλεόρασης [8]. Χρησιμοποιείται στη Γαλλία, Ρωσία, χώρες της πρώην ΕΣΣΔ αλλά και χώρες της κεντρικής Αφρικής. Όπως και τα υπόλοιπα συστήματα, το SECAM πρώτα εφευρέθηκε για κωδικοποίηση ασπρόμαυρης εικόνας και αργότερα τροποποιήθηκε και για τη μετάδοση έγχρωμης εικόνας, με την προϋπόθεση το νέο σύστημα να είναι συμβατό τόσο με τους υπάρχοντες δέκτες ασπρόμαυρης, όσο και με τους νέους έγχρωμης εικόνας. Για αυτό το λόγο εικόνα αποτελείται από δύο σήματα: το σήμα φωτεινότητας (luminance) που κωδικοποιεί την ασπρόμαυρη εικόνα και συμβολίζεται με το γράμμα Y και το χρώμα ή σήμα χρωμικότητας (chrominance) που κωδικοποιεί το χρώμα στην έγχρωμη τηλεόραση και συμβολίζεται με το γράμμα C. Οι δέκτες ασπρόμαυρου σήματος αποκωδικοποιούν μόνο τη φωτεινότητα, ενώ οι δέκτες έγχρωμου αποκωδικοποιούν τόσο τη φωτεινότητα, όσο και το χρώμα.

Για να χωρέσουν και τα δύο σήματα στο εύρος ζώνης μετάδοσης της ασπρόμαυρης εικόνας, το φάσμα του χρώματος επικαλύπτεται με το φάσμα της φωτεινότητας Σχήμα 2.3. Αυτή η επικάλυψη είναι δυνατή γιατί το φάσμα της ασπρόμαυρης εικόνας δεν είναι συνεχές. Οι ψηλότερες συχνότητες, δηλαδή η απότομη αλλαγή από απόλυτο λευκό στο απόλυτο μαύρο από εικονοστοιχείο σε εικονοστοιχείο είναι σπάνια, έτσι μπορούν να αποκοπούν, χωρίς να υπάρχει σημαντική απώλεια στην ποιότητα της εικόνας. Με αυτό τον τρόπο το φάσμα που δεν χρησιμοποιείται για τη μετάδοση ασπρόμαυρης εικόνας, μπορεί να χρησιμοποιηθεί για την κωδικοποίηση του χρώματος.



Σχήμα 2.3: Φάσμα σήματος NTSC

2.4 Σύστημα PAL (Phase Alternating Line)

Εισαγωγή:

Το σύστημα PAL, συντομογραφία του Phase Alternate Line, είναι αναλογικό τηλεοπτικό κωδικοποιητικό σύστημα, όπως το SECAM και το NTSC. Αναπτύχθηκε τη δεκαετία του 50 από χώρες της Δυτικής Ευρώπης με σκοπό να αντικαταστήσει το σύστημα NTSC που παρουσίαζε αδυναμίες, όπως ολίθηση χρώματος κάτω από συνθήκες ασθενούς λήψης τηλεοπτικού σήματος.

Το σύστημα PAL αναπτύχθηκε στα εργαστήρια της Telefunken στη Γερμανία και η πρώτη ματάδοση έγινε το 1963 στο Ηνωμένο Βασίλειο από το τηλεοπτικό σταθμό του BBC. Το σύστημα χρησιμοποιείται από τις περισσότερες χώρες της Ευρώπης, χώρες της Μέσης Ανατολής, Αφρικής αλλά και σε Ινδία, Κίνα και Ωκεανία. Η Ελλάδα, αν και είναι καταχωρημένη διεθνώς ότι χρησιμοποιεί το σύστημα SECAM, όλοι οι τηλεοπτικοί σταθμοί εξέπεμπαν με το σύστημα PAL-B μέχρι το 2012. Με οδηγία της Ευρωπαϊκής Ένωσης από την 1η Ιανουαρίου, οι επίγειες εκπομπές έγιναν ψηφιακές, στο μεγαλύτερο μέρος της χώρας.

Στον Πίνακα 2.1 παραθέτονται τα κύρια χαρακτηριστικά και διαφορές των τριών συστημάτων:

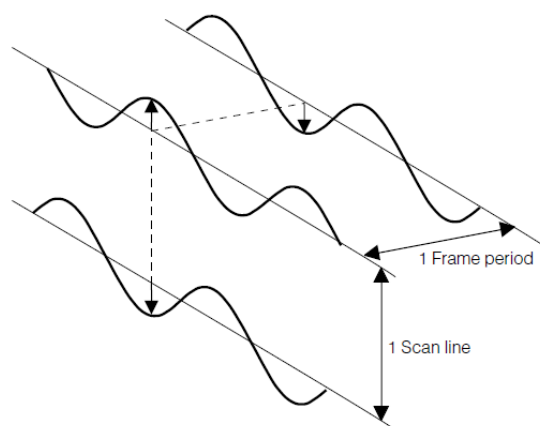
Πίνακας 2.1: Χαρακτηριστικά συστημάτων NTSC, SECAM και PAL [9]

Σύστημα			
Χαρακτηριστικά	NTSC	SECAM	PAL
Οριζόντια Ανάλυση	(αναλογικό)	(αναλογικό)	(αναλογικό)
Κάθετη Ανάλυση	525	625	625
Ορατές Γραμμές ανά Εικόνα	483	610	610
Πεπλεγμένη Σάρωση (interlaced)	Ναι	Ναι	Ναι
Αριθμός Πλαισίων ανά Εικόνα	2	2	2
Συχνότητα Πλαισίων (fields/s)	59.94 Hertz	50 Hertz	50 Hertz
Κωδικοποίηση Φωτεινότητας	AM	FM	AM
Κωδικοποίηση Χρώματος	QAM	QAM	QAM
Κωδικοποίηση Ήχου	FM (F3E)	FM (F3E)	FM (F3E)
Εύρος Ζώνης Καναλιού	6 MHz	8 MHz	7 MHz (VHF) 8 MHz (UHF)

2.5 Αναλυτική Περιγραφή Συστήματος PAL:

Το σύνθετο σήμα (Composite Video):

Το σύνθετο σήμα βίντεο (Composite Video) αρχικά σχεδιάστηκε για τη μετάδοση ασπρόμαυρης εικόνας, στο οποίο αργότερα προστέθηκε η πληροφορία του χρώματος σε διαφορετικό φέρον. Οι απαιτήσεις που έπρεπε να πληροί το νέο σύνθετο σήμα ήταν να επιτρέπει στους δέκτες ασπρόμαυρης και έγχρωμης εικόνας να λαμβάνουν το νέο σήμα χωρίς πρόβλημα και να μην υπάρχει αύξηση στο εύρος ζώνης μετάδοσης του νέου συστήματος. Έτσι το σύνθετο σήμα βίντεο μπορεί να θεωρηθεί σαν μια πρώιμη μορφή συμπίεσης βίντεο [10].



Σχήμα 2.4: Υπέρθεση σημάτων Φωτεινότητας και Χρώματος στο σύνθετο σήμα

Στο Σχ. 2.4 παρουσιάζονται τα σήματα της φωτεινότητας (ευθεία) και χρώματος (ημιτονοειδές). Από την υπέρθεση των δύο σημάτων καταλαβαίνουμε πως το χρώμα αλλοιώνει τη φωτεινότητα τοπικά κάνοντας την εικόνα πιο φωτεινή ή πιο σκοτεινή ανά διαστήματα. Αν η φάση του χρώματος αντιστρεφόταν από γραμμή σε γραμμή, η μέση φωτεινότητα για ένα ζεύγος εικονιστοιχείων θα αντιστοιχούσε στην αρχική φωτεινότητα, και έτσι θα αντιλαμβανόταν και από το ανθρώπινο μάτι. Επίσης αν η φάση του χρώματος αντιστρεφόταν από καρέ σε καρέ, η μέση τιμή της φωτεινότητας για ένα εικονοστοιχείο, στη διάρκεια δύο καρέ, θα αντιστοιχούσε στην επιθυμητή φωτεινότητα. Η συχνότητα του φέροντος πρέπει ωστόσο να επιλεγεί κατάλληλα, έτσι ώστε να έχουμε αυτά τα αποτελέσματα. Τα συστήματα NTSC και PAL διαμορφώνουν τη φάση και το πλάτος του φέροντος έτσι ώστε τα δύο σήματα –φωτεινότητα και χρώμα- να μεταδίδονται ταυτόχρονα, ενώ το SECAM διαμορφώνει κατά συχνότητα το φέρον και μεταδίδει ανά γραμμές είτε τη φωτεινότητα, είτε το χρώμα. Το αποτέλεσμα αυτής της διαμόρφωσης παράγει ένα σήμα με εξαιρετικά πολύπλοκο φάσμα, ειδικά στο σύστημα PAL.

Το πρώτο σύστημα ασπρόμαυρης μετάδοσης τηλεοπτικού σήματος αναπτύχθηκε στις ΗΠΑ το 1940-41. Το 1954 υιοθετήθηκε το σύστημα NTSC-2, δηλαδή το τροποποιημένο σύστημα για μετάδοση έγχρωμης εικόνας. Αυτό το σύστημα εμφάνιζε αστάθειες στην χροιά (hue) του χρώματος λόγω της πολυδιαδρομικής μετάδοσης και των ατελειών στο πομπό. Ένας από τους στόχους της σχεδίασης του συστήματος PAL ήταν να διορθώσει τις αδυναμίες του συστήματος NTSC, ταυτόχρονα επιδιώκοντας να κρατήσει εκτός ευρωπαϊκής αγοράς τους μη-ευρωπαίους κατασκευαστές τηλεοράσεων. Ο δεύτερος στόχος δεν επιτεύχθηκε, αφού Ιάπωνες κατασκευαστές κατάφεραν να σχεδιάσουν αποκωδικοποιητές που συμπεριφέρονταν

στο σήμα σαν να είναι κωδικοποιημένο σε NTSC, αποκωδικοποιώντας μόνο κάθε δεύτερη γραμμή. Η Γαλλία ακολούθησε το δικό της δρόμο, σχεδιάζοντας το σύστημα SECAM, περισσότερο για λόγους εθνικής περηφάνιας, παρά τεχνικούς λόγους. Κάθε χώρα υιοθέτησε το ένα από τα τρία συστήματα, βασιζόμενη περισσότερο στις πολιτικές της σχέσεις, παρά καθαρά τεχνικούς λόγους.

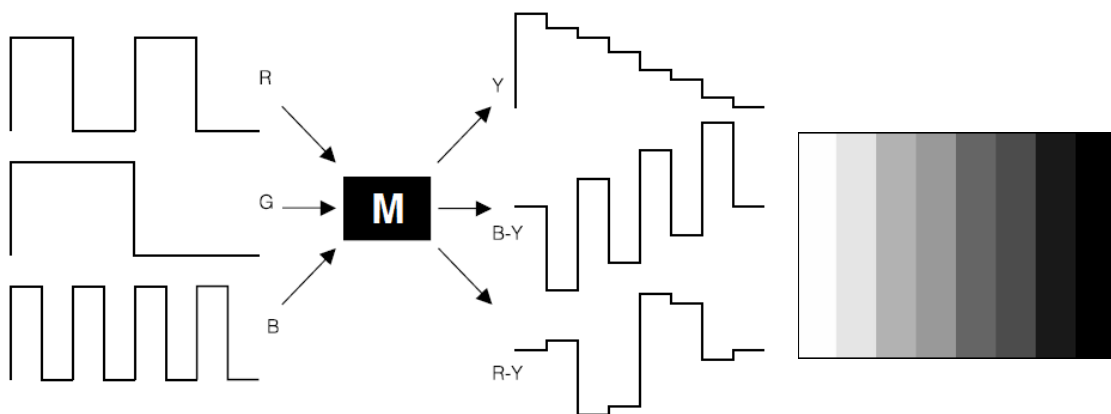
Ο χρωματικός χώρος YUV - Ορθογωνική Διαμόρφωση Πλάτους (Quadrature amplitude modulation – QAM)

Υπάρχουν διάφοροι τρόποι αναπαράστασης μιας εικόνας. [11]. Στο χώρο των υπολογιστών η πιο συνηθισμένη μορφή αναπαράστασης εικόνων είναι στο χρωματικό χώρο RGB (Red, Green, Blue), όπου κάθε εικονοστοιχείο στην εικόνα αναπαριστάται από τρεις μεταβλητές που αντιστοιχούν στο κόκκινο, πράσινο και μπλε χρώμα. Το άθροισμα των τριών αυτών μεταβλητών σχηματίζει το επιθυμητό χρώμα.

Εκτός από το χρωματικό χώρο RGB υπάρχουν και άλλοι χώροι για αναπαράσταση μια εικόνας όπως οι CMYK (Cyan, Magenta, Yellow, Key), YIQ και YUV. Στο παρόν στάδιο παρουσιάζει ενδιαφέρον ο χώρος YUV, αφού αυτός χρησιμοποιείται στη κωδικοποίηση εικόνας στο σύστημα PAL. Παρακάτω παρουσιάζονται οι εξισώσεις μετατροπής από το χώρο RGB στο χώρο YUV:

$$\begin{aligned}
 Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\
 E_U &= 0.493 \cdot (B - Y) \\
 E_V &= 0.877 \cdot (R - Y)
 \end{aligned}$$

Στο Σχ.2.5 παρουσιάζεται πως σχηματίζεται μια από τις πιο γνωστές εικόνες δοκιμής ενός τηλεοπτικού δέκτη. Το κόκκινο (R) εναλλάσσεται κάθε ένα κύκλο, το πράσινο (G) κάθε δύο κύκλους και το μπλε (B) κάθε τέσσερις κύκλους. Όπως φαίνεται και από τις παρακάτω σχέσεις, η φωτεινότητα (Y) σχηματίζει μια σκάλα από το φωτεινότερο (λευκό), στο πιο σκοτεινό (απόλυτο μαύρο) και μπορεί να πάρει μόνο θετικές τιμές.



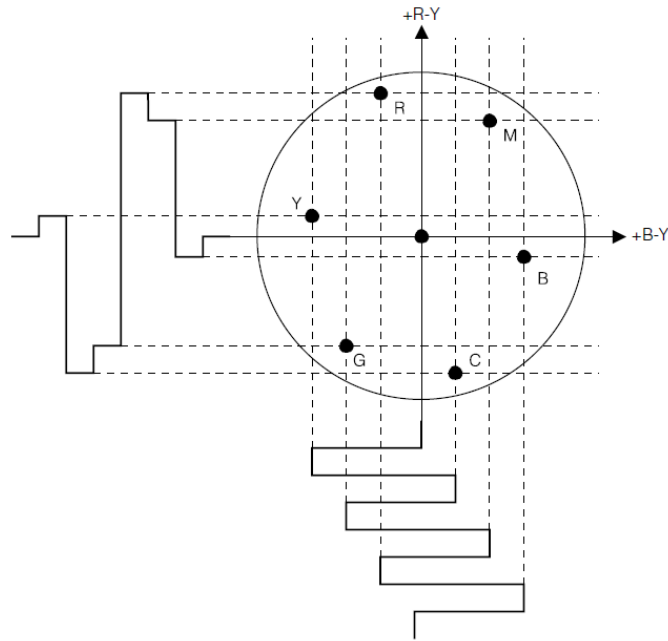
(α)

(β)

Σχήμα 2.5: (α) Σχηματισμός παλέτας δοκιμής στο χρωματικό χώρο YUV, (β) Φωτεινότητα (Κανάλι Y)

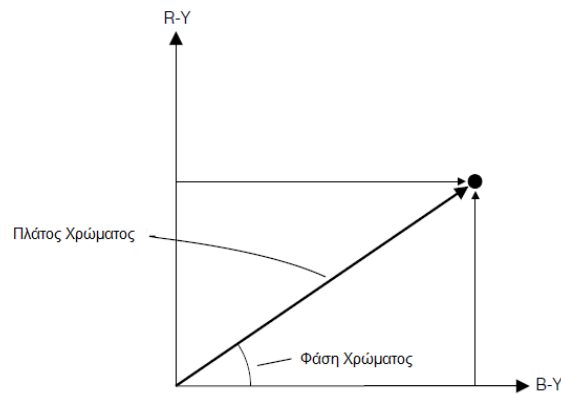
Ακολούθως η φωτεινότητα αφαιρείται από τα χρώματα μπλε και κόκκινο, δημιουργώντας τα διανύσματα E_U και E_V ή απλώς U και V όπως ονομάζονται συνήθως. Τα χρώματα μπορούν να παραστηθούν γραφικά, σε ένα διάγραμμα με τα διανύσματα R-Y και B-Y στους δύο

άξονες, για σταθερή τιμή της φωτεινότητας. Όπως παρατηρούμε, το λευκό και μαύρο εμφανίζονται στην αρχή των αξόνων. Όσο απομακρυνόμαστε από το κέντρο των αξόνων αυξάνεται ο κορεσμός των χρωμάτων, ενώ τα χρώματα παρουσιάζονται σε διάφορες θέσεις του διαγράμματος.

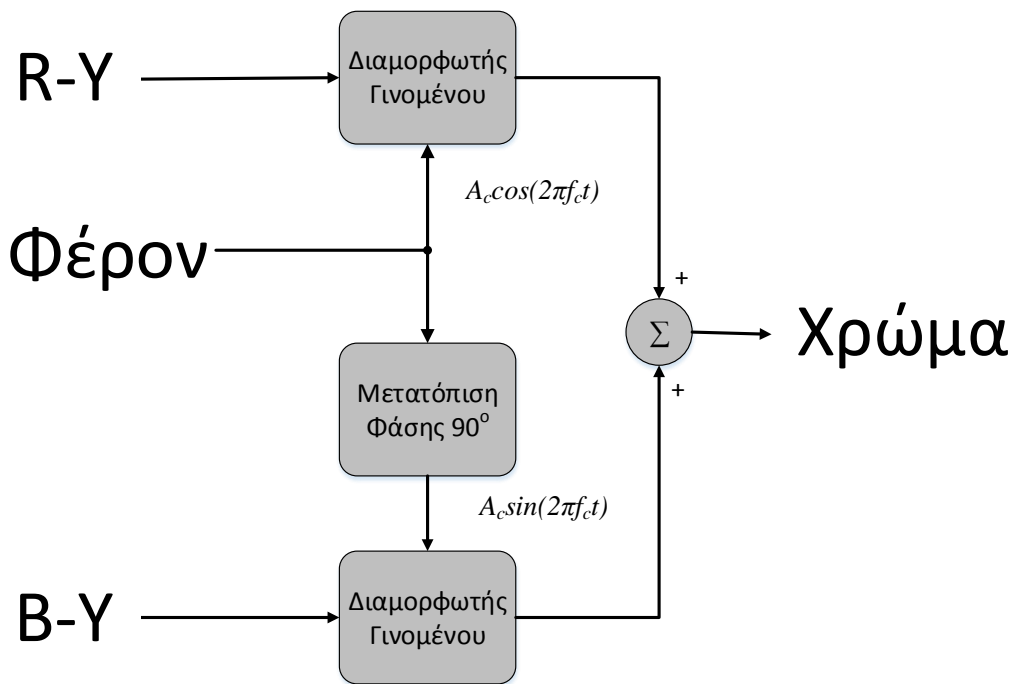


Σχήμα 2.6: Αναπαράσταση των διανυσμάτων R-Y και B-Y

Εναλλακτικά αντί να χρησιμοποιήσουμε δύο σήματα, μπορούμε να αναπαραστήσουμε ένα σημείο στο χώρο χρωμάτων YUV με το πλάτος και φάση μιας κυματομορφής. Αυτός είναι ακριβώς ο τρόπος με τον οποίο γίνεται η κωδικοποίηση του χρώματος στα συστήματα PAL και NTSC. Η ακτίνα ενός σημείου στην χρωματική παλέτα αντιστοιχεί στο πλάτος που είναι ανάλογο του κορεσμού (saturation) και η γωνία με τη φάση, ανάλογη της απόχρωσης (hue). Αυτή η ταυτόχρονη μετάδοση των δύο σημάτων γίνεται με την ορθογωνική διαμόρφωση πλάτους (QAM) Σχ. 2.8.



Σχήμα 2.7: Αναπαράσταση χρώματος με πλάτος και φάση

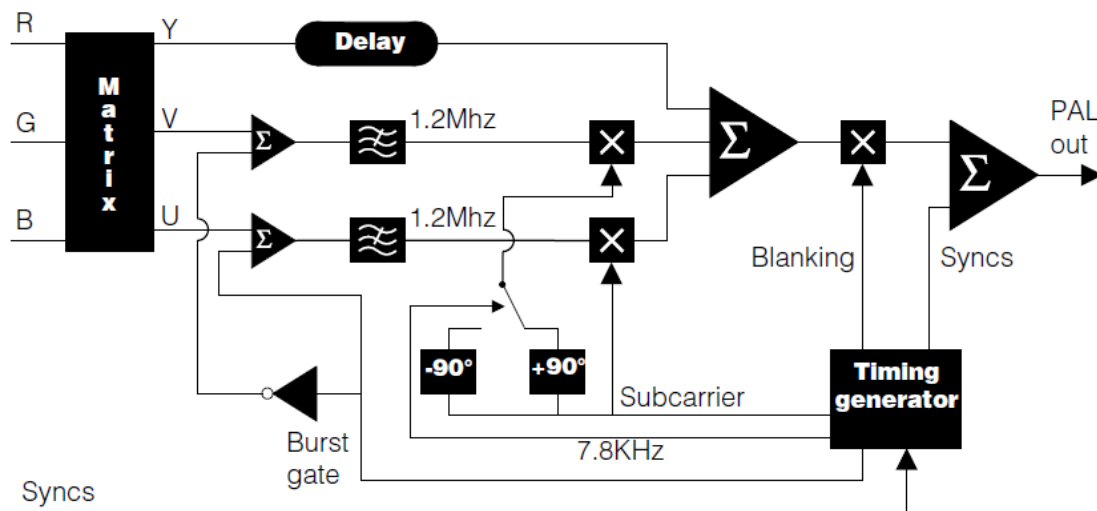


Σχήμα 2.8: Ορθογωνικός Διαμορφωτής Πλάτους

Ένα ζεύγος από διαμορφωτές γινομένου διαμορφώνει το ίδιο φέρον, με τη διαφορά ότι στον ένα, έχει μετατοπιστεί η φάση κατά 90° . Η έξοδος των δύο διαμορφωτών γινομένου προστίθεται γραμμικά, έτσι το προκύπτον σήμα είναι διαμορφωμένο τόσο κατά πλάτος, όσο και κατά φάση. Η έξοδος προκύπτει με καταπιεσμένο φέρον. Για την αποδιαμόρφωση QAM, χρειάζεται αποδιαμορφωτής που να γνωρίζει τη φάση των δύο αρχικών φερόντων. Έτσι μαζί με το σήμα πρέπει να μεταδίδεται αναγκαστικά και στιγμιότυπο του φέροντος σε ριπές (bursts). Οι ριπές μεταδίδονται κατά τη διάρκεια των παλμών οριζώντιου συγχρονισμού.

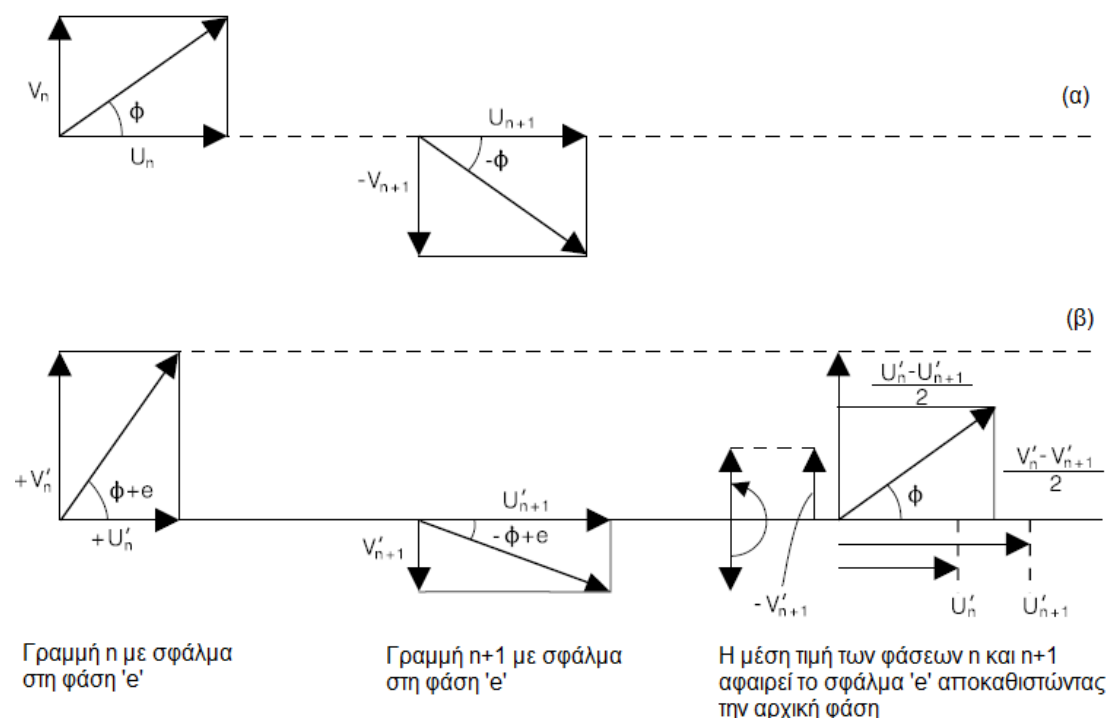
Το σύστημα PAL:

Το σύστημα PAL (Phase Alternating Line) σχεδιάστηκε με σκοπό να αντιμετωπίσει την ευαισθησία σε σφάλματα της απόχρωσης και να απαλλάξει τους τηλεοπτικούς δέκτες από τα κυκλώματα διόρθωσης της απόχρωσης που ήταν απαραίτητα στο σύστημα NTSC. Στο Σχήμα 2.9 βλέπουμε ότι η είσοδος RGB μετατρέπεται στα διανύσματα Y, B-Y, R-Y που αργότερα πολλαπλασιαζόμενα με τα αντίστοιχα βάρη δίνουν τα σήματα Y,U και V.



Σχήμα 2.9: Διάγραμμα κωδικοποιητή PAL

Η μετάδοση του χρώματος στο σύστημα PAL γίνεται με ορθογωνική διαμόρφωση πλάτους, αλλά σε κάθε γραμμή η φάση του διανύσματος V αντιστρέφεται.



Σχήμα 2.10: Μηχανισμός αφαίρεσης σφαλμάτων φάσης στο σύστημα PAL
 (α) Εκπεμπόμενο σήμα
 (β) Λαμβανόμενο σήμα

Στο Σχήμα 2.10 βλέπουμε πώς η εναλλαγή της φάσης του διανύσματος V απαλλάσσει το λαμβανόμενο σήμα από σφάλματα στη φάση. Στο παράδειγμα βλέπουμε πως το σφάλμα στη φάση 'e' προκαλεί αύξηση στο διάνυσμα V και μείωση στο διάνυσμα U στη γραμμή n, ενώ προκαλεί το αντίθετο αποτέλεσμα στην επόμενη γραμμή. Σε διάστημα δύο συνεχών γραμμών παρατηρούμε ότι η μέση τιμή των διανυσμάτων U και V προκύπτει ίση με την αρχική, έτσι αποτρέπονται οι αλλοιώσεις στην απόχρωση που αντιμετώπιζαν οι δέκτες NTSC.

Στους απλούς δέκτες PAL, η ομογενοποίηση του σήματος από γραμμή σε γραμμή θεωρείται ότι γίνεται από το θεατή, και όχι από ειδικό κύκλωμα, έτσι μεγάλα σφάλματα στη φάση γίνονται αντιληπτά σαν μεταβολές στη φωτεινότητα μεταξύ δύο διαδοχικών γραμμών και καλούνται γραμμές Hanover (Hanover Bars) Σχ. 2.11. Για αντιμετώπιση αυτού του φαινομένου, σχεδιάστηκαν δέκτες που χρησιμοποιούν γραμμή καθυστέρησης (PAL-D - delay) που αποθηκεύει τη πληροφορία χρώματος για κάθε γραμμή και ακολούθως την αθροίζει με την επόμενη πριν οδηγηθεί στην έξοδο. Αυτή η μέθοδος μπορεί να διορθώσει μεγάλα σφάλματα στη φάση, όμως μειώνει στο μισό την κάθετη ανάλυση του χρώματος της εικόνας.



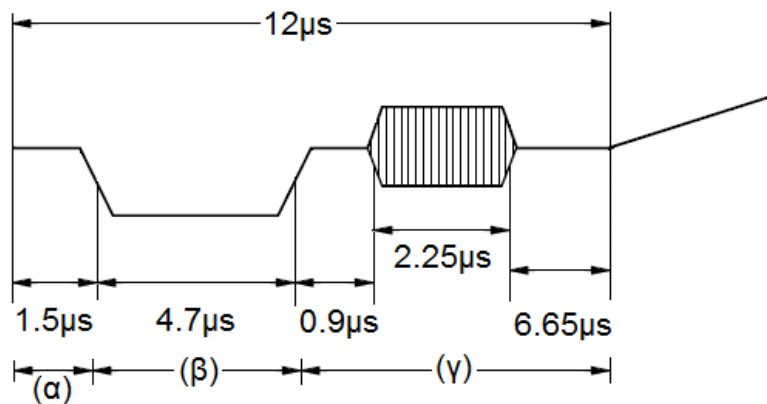
Σχήμα 2.11: Αποτελέσματα μεγάλων σφαλμάτων στη φάση (Hanover Bars)

Επιστρέφοντας στο Σχ.2.9 βλέπουμε πως η εναλλαγή της φάσης του διανύσματος V γίνεται αντιστρέφοντας τη φάση του φέροντος κατά $\pm 90^\circ$ σε σχέση με το U . Η αντιστροφή των ριπών φέροντος επιτυγχάνεται προσθέτοντας στο διάνυσμα U το φέρον και στο διάνυσμα V το αντεστραμμένο φέρον. Το σήμα φωτεινότητας περνά από ένα δίκτυο καθυστέρησης πριν προστεθεί με τα διανύσματα U , V και τους παλμούς συγχρονισμού. Για τη μετάδοση το σύνθετο σήμα έχει συνολικό εύρος ζώνης 5.75 MHz.

Για το σχηματισμό μιας εικόνας στην οθόνη του τηλεοπτικού δέκτη, μεταδίδονται 625 γραμμές Σχ. 2.13. Από αυτές μόνο οι 610 περιέχουν πληροφορία που εμφανίζεται στην οθόνη, ενώ οι υπόλοιπες στέλνουν σήματα συγχρονισμού για οριζόντιο και κάθετο συγχρονισμό και περιέχουν πληροφορία για τηλεκείμενο (teletext).

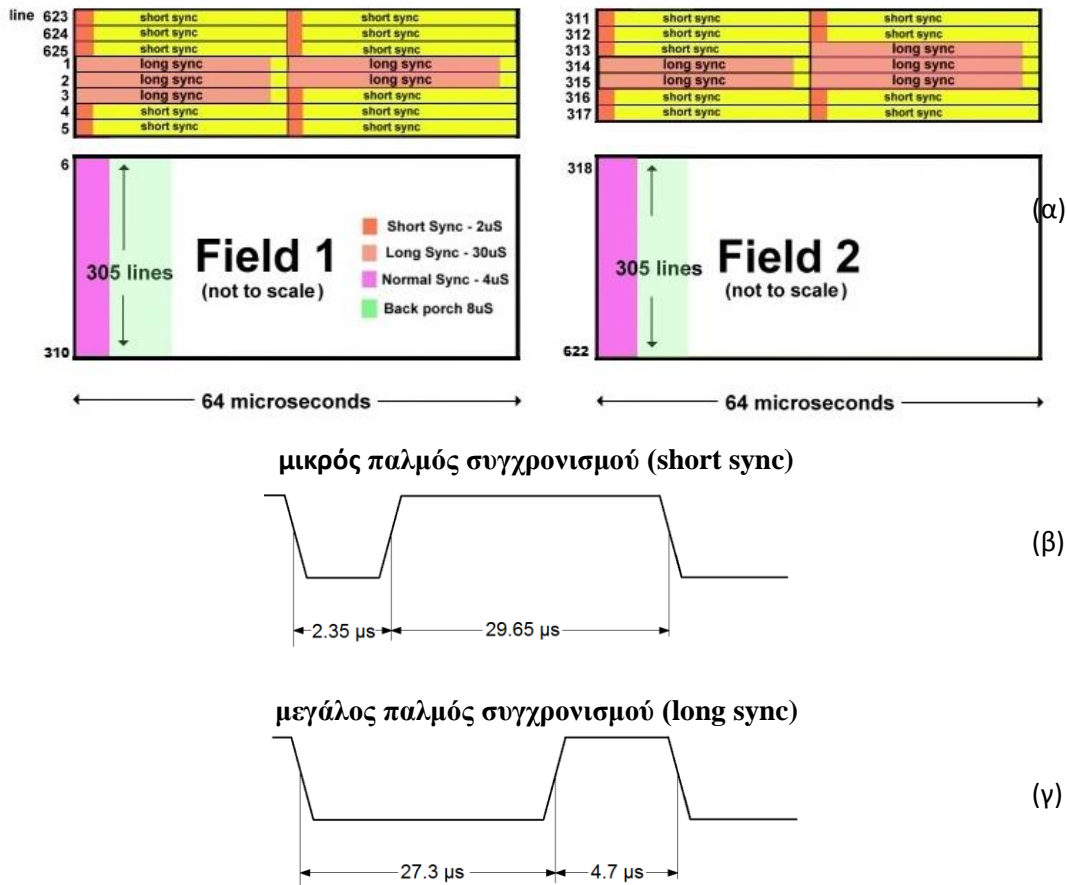
Μεταδίδονται 50 πλαίσια ανά δευτερόλεπτο, που κάθε πλαίσιο περιέχει τη μισή πληροφορία για το σχηματισμό μιας εικόνας. Το πλαίσιο 1 μεταδίδει τη πληροφορία για τις περιττές γραμμές και το πλαίσιο 2 μεταδίδει τη πληροφορία για τις άρτιες γραμμές στην τηλεόραση. Οπότε κάθε χρονική στιγμή στην οθόνη της τηλεόρασης εμφανίζεται μόνο η μισή εικόνα. Μέσω του μηχανισμού επιμονής της όρασης (persistence of vision), ο ανθρώπινος εγκέφαλος δεν αντιλαμβάνεται αυτή τη ταχεία αλλαγή και βλέπει ολόκληρη την εικόνα. Αφού μεταδίδονται 2 πλαίσια με συχνότητα 50 Hz, ο ρυθμός μετάδοσης του συστήματος PAL είναι 25 καρέ ανά δευτερόλεπτο.

Κάθε μεταδιδόμενη γραμμή έχει διάρκεια 64 μ s, 52 από τα οποία περιέχουν την εικόνα που μεταδίδεται στην οθόνη, ενώ τα πρώτα 12 μ s περιέχουν παλμούς οριζόντιου συγχρονισμού και ριπές φέροντος χρώματος. Συγκεκριμένα τα πρώτα 1.5 μ s ονομάζονται μπροστινό μέρος (Front Porch) και ακολουθούνται από ένα παλμό οριζόντιου συγχρονισμού διάρκειας 4.7 μ s που οδηγεί το σήμα από τα 0,3 Volt στα 0 Volt. Ακολουθεί το πίσω μέρος (Back Porch) στο οποίο μεταδίδεται η ριπή φέροντος χρώματος διάρκειας 2.25 μ s Σχ.2.12. Να τονιστεί ότι ακόμα και αν δεν μεταδίδεται η ριπή φέροντος, οι χρονικοί των τριών μερών πρέπει να παραμένουν όπως φαίνεται στο σχήμα.



Σχήμα 2.12: Μη ορατό μέρος σε μια γραμμή εικόνας
 (α) Μπροστινό μέρος (Front Porch)
 (β) Παλμός οριζόντιου συγχρονισμού
 (γ) Πίσω μέρος (Back Porch)

Εκτός από τις ορατές γραμμές, μεταδίδονται και γραμμές κάθετου συγχρονισμού για να σηματοδοτήσουν την έλευση κάθε πλαισίου εικόνας. Για το πλαίσιο 1 οι γραμμές συγχρονισμού ξεκινούν από τη γραμμή 623 και τελιώνουν στη γραμμή 5. Αποτελούνται από έξι παλμούς προ-ισοστάθμισης (pre-equalizing pulses), πέντε μακριούς παλμούς συγχρονισμού και πέντε παλμούς μετα-ισοστάθμισης (post-equalizing pulses). Για το πλαίσιο 2 οι παλμοί συγχρονισμού ξεκινούν από τη γραμμή 311 και τελιώνουν στη γραμμή 317. Αποτελούνται από πέντε παλμούς προ-ισοστάθμισης (pre-equalizing pulses), πέντε μακριούς παλμούς συγχρονισμού και τέσσερις παλμούς μετα-ισοστάθμισης (post-equalizing pulses). Στο Σχ.2.13 παρουσιάζεται η δομή για τη μετάδοση μιας εικόνας με κωδικοποίηση PAL.



Σχήμα 2.13: (α) Δομή κατά τη μετάδοση μιας εικόνας στο σύστημα PAL
 (β) Μικρός παλμός συγχρονισμού
 (γ) Μεγάλος παλμός συγχρονισμού

Το πλάτος του σύνθετου σήματος (composite video) παίρνει τιμές 0-1 Volt, με τη στάθμη των 0 Volt να θεωρείται στάθμη συγχρονισμού, τα 0,3 Volt να αντιστοιχούν στο απόλυτο μαύρο και τα 1 Volt να αντιστοιχούν στο απόλυτο λευκό.

Η μετάδοση του τηλεοπτικού σήματος γίνεται σε κανάλια στις περιοχές των VHF (30 MHz με 300 MHz) και UHF (300 MHz με 3 GHz). Το εύρος ζώνης κάθε καναλιού είναι 7 MHz για τα VHF και 8 MHz για τα UHF. Παρακάτω παρατίθενται τα κανάλια εκπομπής για τις δύο περιοχές συχνοτήτων, για την Ελλάδα.

VHF

Κανάλι	Εύρος Ζώνης (MHz)	Συχνότητα Φέροντος Βίντεο (MHz)	Συχνότητα Φέροντος Ήχου (MHz)
2	47-54	48.25	53.75
3	54-61	55.25	60.75
4	61-68	62.25	67.75
5	174-181	175.25	180.75
6	181-188	182.25	187.75
7	188-195	189.25	194.75
8	195-202	196.25	201.75
9	202-209	203.25	208.75
10	209-216	210.25	215.75
11	216-223	217.25	222.75
12	223-230	224.25	229.75

UHF

Κανάλι	Εύρος Ζώνης (MHz)	Συχνότητα Φέροντος Βίντεο (MHz)	Συχνότητα Φέροντος Ήχου (MHz)
21	470-478	471.25	476.75
22	478-486	479.25	484.75
23	486-494	487.25	492.75
24	494-502	495.25	500.75
25	502-510	303.25	508.75
26	510-518	511.25	516.75
27	518-526	519.25	524.75
28	526-534	527.25	532.75
29	534-542	535.25	540.75
30	542-550	543.25	548.75
31	550-558	551.25	556.75
32	558-566	559.25	564.75
33	566-574	567.25	572.75
34	574-582	575.25	580.75
35	582-590	583.25	588.75
36	590-598	591.25	596.75
37	598-606	599.25	604.75
38	606-614	607.25	612.75
39	614-622	615.25	620.75
40	622-630	623.25	628.75

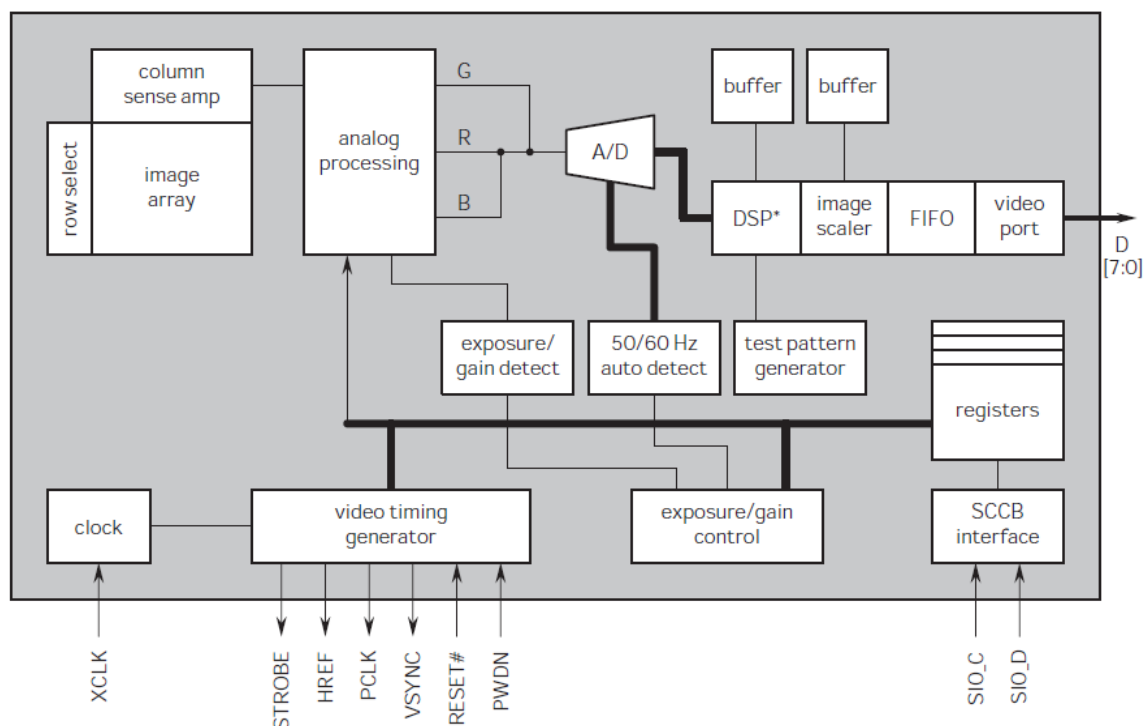
Σχήμα 2.14: Κανάλια εκπομπής αναλογικής τηλεόρασης για Ελλάδα

2.6 Το πρωτόκολλο VGA

Το αναλογικό πρωτόκολλο VGA αναπτύχθηκε από την IBM το 1987. Αν και αρχικά προοριζόταν για χρήση στους προσωπικούς ηλεκτρονικούς υπολογιστές, τα τελευταία χρόνια έχει μεταπηδήσει και σε άλλες περιοχές, όπως οι φορητές ηλεκτρονικές συσκευές. Για τη μετάδοση του σήματος VGA, χρησιμοποιούνται 5 κανάλια, τα *vsync*, *hsync*, *red*, *green*, και *blue*. Το σήμα *hsync* σηματοδοτεί το τέλος μετάδοσης μιας γραμμής, ενώ το *vsync* το τέλος μετάδοσης ολόκληρης της εικόνας. Σε κάθε γραμμή μεταδίδονται 640 εικονοστοιχεία σε μορφή RGB, ενώ συνολικά μεταδίδονται 480 γραμμές.

2.7 Η κάμερα OV7670

Για την υλοποίηση του πομπού σε FPGA, χρησιμοποιήσαμε την VGA κάμερα OV7670 της εταιρίας Omnivision [12].



Σχήμα 2.15: Λειτουργικό διάγραμμα βιντεοκάμερας OV7670

Η κάμερα έχει πέντε ακροδέκτες εισόδου, τους XCLK, RESET#, PWDN, SOIC, SOID και ένδεκα ακροδέκτες εξόδου, τους STROBE, HREF, PCLK, VSYNC και το διαύλο 8-bit D. Παρακάτω αναλύονται οι ακροδέκτες εισόδου/εξόδου και οι λειτουργίες τους:

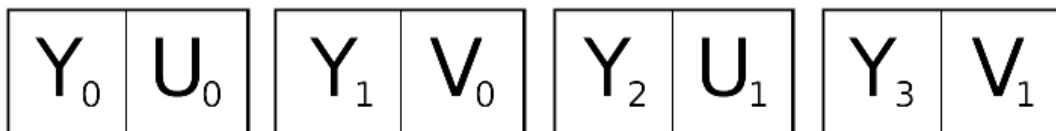
Ακροδέκτης	Τύπος ακροδέκτη	Λειτουργία
XCLK	Είσοδος	Σήμα ρολογιού αναφοράς εισόδου
RESET#	Είσοδος	Επαναφέρει όλους τους καταχωρητές στις αρχικές τους τιμές. Ανάστροφη λογική: 0: Επαναφορά, 1: Κανονική λειτουργία

PWDN	Είσοδος	Απενεργοποίηση βιντεοκάμερας. 0: Κανονική λειτουργία 1: Απενεργοποίηση
SOIC	Είσοδος	Σειριακός Δίαυλος Ελέγχου Βιντεοκάμερας (Serial Camera Control Bus), για εγγραφή και ανάγνωση από τους καταχωρητές της κάμερας
SOID	Είσοδος	
STROBE	Έξοδος	Για στροβοσκοπικό έλεγχο LED ή Flash
HREF	Έξοδος	Σήμα οριζόντιου συγχρονισμού
PCLK	Έξοδος	Έξοδος σήματος ρολογιού εικονοστοιχείου
VSYNC	Έξοδος	Σήμα κάθετου συγχρονισμού
D[7:0]	Έξοδος	Έξοδος εικόνας πλάτους 8-bit. Μπορεί να μεταδώσει εικονοστοιχεία σε χρωματικούς κώδικες YUV, RGB και RAW.

Από τους παραπάνω ακροδέκτες χρησιμοποιούνται οι ακροδέκτες εισόδου XCLK, RESET#, PWDN και εξόδου HREF, PCLK, VSYNC, D ενώ οι ακροδέκτες εισόδου SOIC και SOID που δεν χρησιμοποιούνται οδηγούνται στα 0 Volt.

Μετάδοση YUV 4:2:2

Στην παρούσα υλοποίηση κρίθηκε σκόπιμο η μετάδοση της εικόνας από τη κάμερα να γίνεται στη μορφή YUV 4:2:2. Σε αυτή τη μορφή, για κάθε εικονοστοιχείο μεταδίδονται δύο byte. Στο πρώτο byte μεταδίδεται η φωτεινότητα Y, ενώ στο δεύτερο byte μεταδίδονται εναλλάξ είτε το χρώμα U, είτε το χρώμα V. Έτσι αν θέλουμε να μεταδώσουμε ασπρόμαυρη εικόνα, μπορούμε μόνο να χρησιμοποιήσουμε μόνο το πρώτο από τα δύο byte.



3

Εισαγωγή στη τεχνολογία FPGA

3.1 Εισαγωγή

Σε αυτό το εισαγωγικό κεφάλαιο, επιχειρείται μια σύντομη εισαγωγή και περιγραφή της συστοιχίας επιτόπια προγραμματιζόμενων πυλών (Field Programmable Gate Arrays - FPGA). Αφού γίνει μια σύντομη ιστορική αναδρομή, στις συστοιχίες επιτόπια προγραμματιζόμενων πυλών, ακολουθεί η περιγραφή της αρχιτεκτονικής των FPGA. Ακολούθως αναλύεται η διαδικασία σχεδίασης και υλοποίησης ενός συστήματος σε FPGA και παρουσιάζονται τα κύρια χαρακτηριστικά της οικογένειας Spartan 3E της εταιρίας Xilinx όπου έγινε και η υλοποίηση της κατασκευής.

3.2 Συστοιχίες επιτόπια προγραμματιζόμενων πυλών (FPGAs)

Ένα κύκλωμα συστοιχίας επιτόπια προγραμματιζόμενων πυλών (FPGA) είναι μια συσκευή ημιαγωγών αποτελούμενη από ένα πλέγμα προγραμματιζόμενων λογικών μπλοκ (Configurable Logic Blocks - CLBs) που συνδέεται μέσω προγραμματιζόμενων διασυνδέσεων. Το FPGA μπορεί να επαναπρογραμματιστεί έτσι ώστε να εκτελεί την επιθυμητή λειτουργία μετά την κατασκευή του. Αυτό το χαρακτηριστικό διαφοροποιεί τα FPGA από τα ολοκληρωμένα κυκλώματα ειδικού σκοπού (ASIC) που κατασκευάζονται έτσι ώστε να εκτελούν ειδικές λειτουργίες και απαιτήσεις. Αν και στο εμπόριο διατίθενται και FPGA που προγραμματίζονται μόνο μια φορά, χωρίς δυνατότητα επαναπρογραμματισμού, η πλειονότητα των FPGA είναι επαναπρογραμματιζόμενα, βασισμένα σε SRAM και παρέχουν τη δυνατότητα στο χρήστη να τα επαναπρογραμματίζει, ενώ εξελίσσεται ο σχεδιασμός του συστήματος.

Τα FPGA είναι ιδανικά για εφαρμογές που οι παράμετροι λειτουργίας πρέπει να αλλάζουν συχνά ή για μικρές ποσότητες παραγωγής, ενώ τα ASIC λόγω μαζικής παραγωγής είναι φτηνότερα σε περιπτώσεις που απαιτούνται μεγάλες ποσότητες και όπου η λειτουργία είναι αυστηρά καθορισμένη και χωρίς αλλαγές. Από την άλλη τα FPGAs δεν έχουν τη ταχύτητα που προσφέρουν τα ASIC και καταναλώνουν αρκετή ενέργεια λόγω της μορφής των διασυνδέσεών τους. Έτσι δεν ενδείκνυνται για εφαρμογές υψηλής ταχύτητας, ούτε για φορητές συσκευές, όπου η χαμηλή κατανάλωση ενέργειας είναι υψίστης σημασίας.

3.3 Ιστορική Αναδρομή

Η τεχνολογία FPGA έχει τις ρίζες τις στις προγραμματιζόμενες μνήμες μόνο για ανάγνωση (PROM) και στις προγραμματιζόμενες λογικές διατάξεις (PLDs). Τόσο οι PROM, όσο και οι PLD, είχαν τη δυνατότητα να προγραμματιστούν στο χώρο κατασκευής τους ή στο πεδίο λειτουργίας (field programmable), ωστόσο οι λογικές πύλες ήταν μόνιμα συνδεδεμένες με την προγραμματιζόμενη λογική.

Στα τέλη της δεκαετίας του 1980 το Πολεμικό Ναυτικό των ΗΠΑ χρηματοδότησε την ιδέα του Steve Casselman για σχεδίαση ενός υπολογιστή αποτελούμενου από 600,000 επαναπρογραμματίσιμες πύλες. Το εγχείρημα του Casselman στέφθηκε με επιτυχία και καταχωρήθηκε σαν ευρεσιτεχνία το 1992.

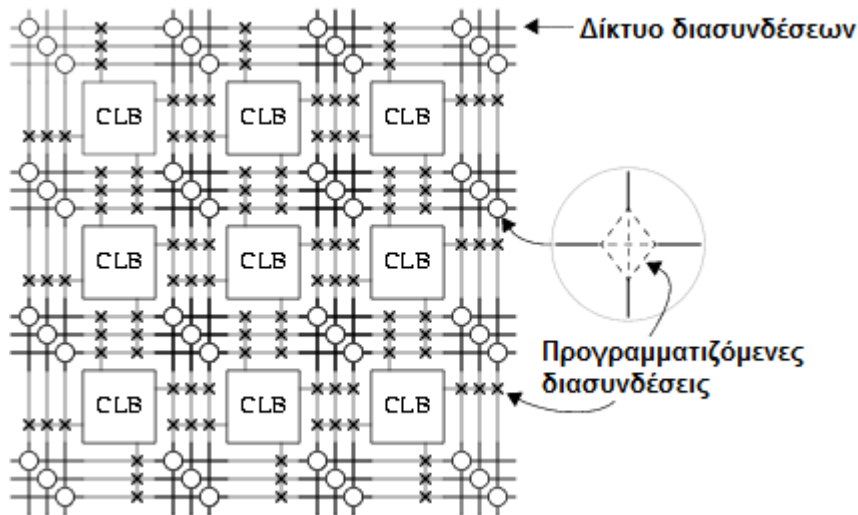
Η πρώτη εμπορικά διαθέσιμη συστοιχία επαναπρογραμματιζόμενων λογικών πυλών, η XC2064, έκανε την εμφάνισή της από τους Ross Freeman και Bernard Vonderschmitt το 1985. Το SC2064 είχε 64 προγραμματιζόμενα λογικά μπλοκ που περιείχαν LUTs τριών εισόδων και συνδέονταν μεταξύ τους μέσω προγραμματιζόμενων διασυνδέσεων. Είκοσι χρόνια αργότερα ο Freeman συνιδρυτής της εταιρίας Xilinx θα έμπαινε στο National Inventors Hall of Fame για την εφεύρεσή του.

Η Xilinx συνέχισε ακάθεκτη, χωρίς ανταγωνισμό από το 1985 έως τα μέσα της δεκαετίας του 90, οπότεν εμφανίστηκαν ανταγωνιστικές εταιρίες, διεκδικώντας σημαντικό μερίδιο της αγοράς FPGA. Εώς το 1993 η Actel, μια ανταγωνιστική εταιρία, είχε καταφέρει να έχει μερίδιο 18% στην αγορά των FPGA.

Η δεκαετία του 1990 ήταν μια εκρηκτική περίοδος για την τεχνολογία των FPGA, τόσο για την αύξηση της πολυπλοκότητας, όσο και για την αύξηση των πωλήσεων. Από τη χρήση στις τηλεπικοινωνίες και στα δίκτυα, η τεχνολογία FPGA σιγά σιγά κατέκτησε και άλλους τομείς όπως η αυτοκινητοβιομηχανία, οι βιομηχανικές εφαρμογές αλλά και άλλα καταναλωτικά προϊόντα.

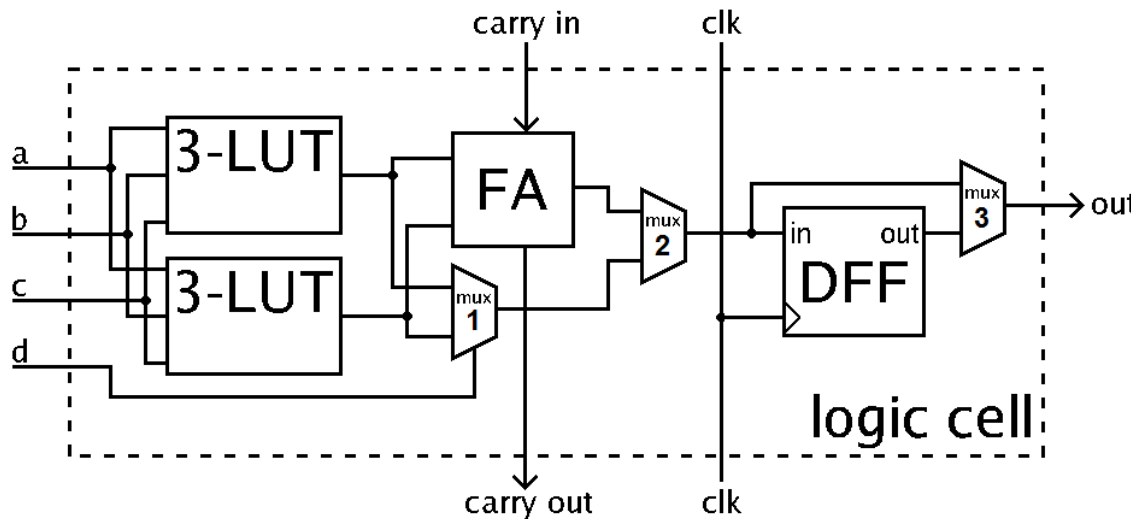
3.4 Αρχιτεκτονική FPGA

Η κλασσική αρχιτεκτονική των FPGA αποτελείται από μια συστοιχία προγραμματιζόμενων λογικών μπλοκ (Configurable Logic Block - Xilinx ή Logic Array Block - Altera), κανάλια διασύνδεσης και ακίδες εισόδου/εξόδου. Κάθε κανάλι διασύνδεσης έχει το ίδιο πλάτος (αριθμό γραμμών) ενώ πολλαπλές ακίδες εισόδου/εξόδου είναι δυνατόν να χωρέσουν στο ύψος μιας γραμμής ή το πλάτος μιας στήλης. Στην παρακάτω εικόνα παρουσιάζεται ένα απλοποιημένο σχέδιο της αρχιτεκτονικής των FPGA [13].



Σχήμα 3.1: Απλοποιημένο μοντέλο αρχιτεκτονικής FPGA

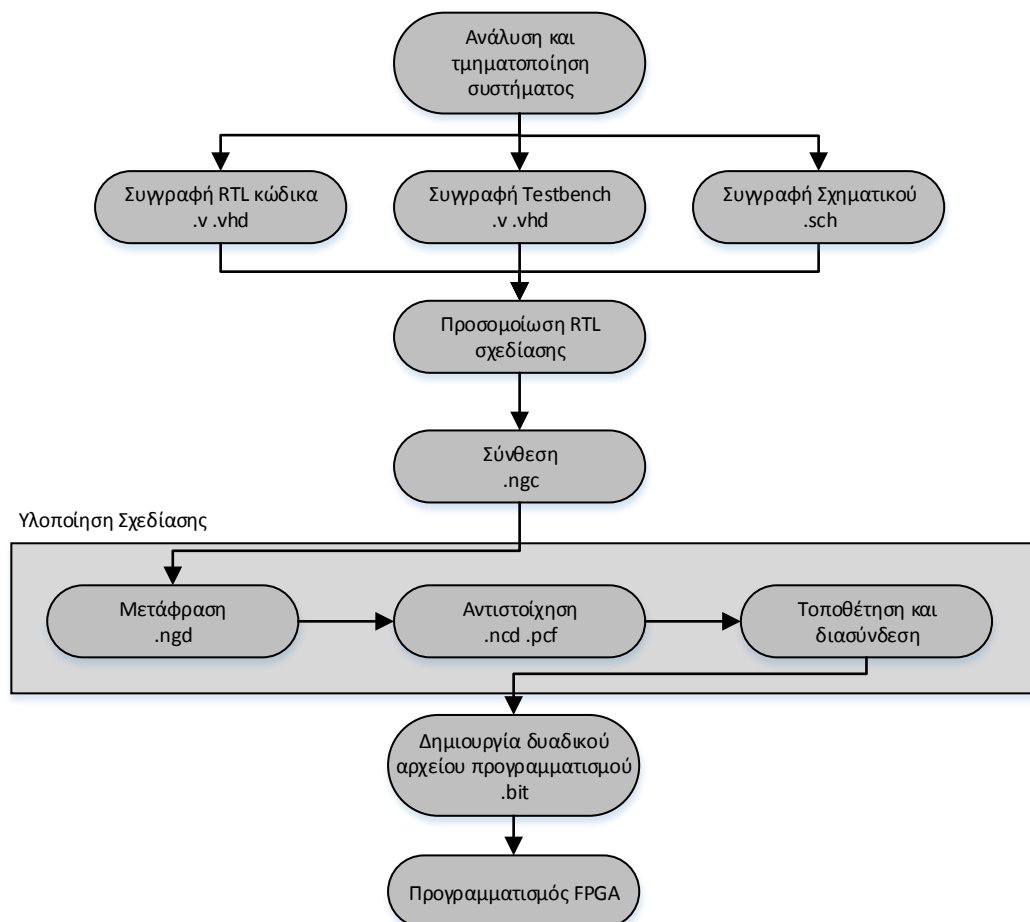
Κάθε CLB αποτελείται από ένα LUT 4 εισόδων, ένα πλήρη αθροιστή (FA), ένα D flip-flop και πολυπλέκτες. Όπως φαίνεται και στο Σχ.3.2 ο LUT 4 εισόδων αποτελείται από δύο LUT των 3 εισόδων. Σε κανονική λειτουργία (*normal mode*), οι 2 LUT λειτουργούν σαν ένας LUT 4 εισόδων και οδηγούνται μέσω του πολυπλέκτη 1. Σε αριθμητική λειτουργία (*arithmetic mode*) οι εξόδοι των δύο LUT οδηγούνται στον FA. Η επιλογή της λειτουργίας επιλέγεται μέσω του πολυπλέκτη 2. Η έξοδος του CLB μπορεί να είναι σύγχρονη ή ασύγχρονη, αν οδηγείται μέσω του D flip-flop και προγραμματίζεται μέσω του πολυπλέκτη 3. Στην πραγματικότητα ολόκληρος ή μέρος του FA υλοποιείται σαν συνάρτηση των LUTs με σκοπό να εξοικονομηθεί χώρος.



Σχήμα 3.2: Απλοποιημένο μοντέλο αρχιτεκτονικής FPGA

3.5 Διαδικασία Σχεδίασης/ Υλοποίησης προγράμματος σε FPGA

Για τη σχεδίαση ενός συστήματος σε υλικό ή τον προγραμματισμό ενός FPGA, ο σχεδιαστής περιγράφει το υλικό προς σχεδίαση με μια γλώσσα περιγραφής υλικού (hardware description language, hdl) ή με ένα σχηματικό διάγραμμα. Οι πιο γνωστές γλώσσες περιγραφής υλικού είναι οι VHDL (Very-high-speed integrated circuit Hardware Description Language) και Verilog. Αφού γίνει η σχεδίαση του συστήματος σε επίπεδο καταχωρητών (Register Transfer Level, RTL) ακολουθούν προσομοιώσεις για να ελεγχθεί η σωστή λειτουργία του κυκλώματος σε επίπεδο rtl. Για αυτό τον έλεγχο γίνεται χρήση ενός ηλεκτρονικού εργαλείου σχεδίασης (Electronic Design Automation tool, EDA) και διαφόρων testbenches που δίνουν τις απαραίτητες εισόδους για έλεγχο του συστήματος. Στη συνέχεια με τη διαδικασία της σύνθεσης η σχεδίαση μετατρέπεται σε ένα netlist αρχείο (.ngc). Ακολούθως το αρχείο netlist μετατρέπεται σε ένα native generic database (.ngd) αρχείο, δηλαδή σε αρχικές δομές της κάθε τεχνολογίας. Αυτή η διαδικασία ονομάζεται μετάφραση. Οι αρχικές δομές αντιστοιχούνται σε συγκεκριμένα CLBs και IOBs μέσω της διαδικασίας αντιστοίχισης (mapping). Τέλος ακολουθεί η διαδικασία τοποθέτησης και διασύνδεσης (place & route) που παράγει το δυαδικό κώδικα (bitstream) για τον προγραμματισμό του συγκεκριμένου FPGA. Ο προγραμματισμός των FPGA γίνεται μέσω ειδικού καλωδίου προγραμματισμού και ελέγχου JTAG ή μέσω της θύρας USB που διαθέτουν οι περισσότερες αναπτυξιακές πλακέτες FPGA. Στο Σχ. 3.3 παρουσιάζεται η διαδικασία σχεδίασης και υλοποίησης ενός προγράμματος σε FPGA [14].



Σχήμα 3.3: Διαδικασία σχεδίασης και υλοποίησης κυκλώματος σε τεχνολογία FPGA

Αρκετές φορές όλα τα εργαλεία για υλοποίηση ενός συστήματος από την αρχή μέχρι το τέλος περιλαμβάνονται σε ένα ενιαίο περιβάλλον. Η εταιρία Xilinx για αυτό το σκοπό παρέχει το Xilinx ISE (Integrated Software Environment) που περιέχει όλα τα εργαλεία για την συγγραφή και προσημείωση του κώδικα αλλά και τη μετάφραση και προγραμματισμό του FPGA.

Τέλος για την ευκολία στη σχεδίαση σύνθετων συστημάτων, υπάρχουν βιβλιοθήκες που υλοποιούν σύνθετες δομές, που είναι ελεγμένες για τη σωστή και αποδοτική λειτουργία τους. Αυτές οι έτοιμες δομές ή IP Cores όπως ονομάζονται προσφέρονται από τις εταιρίες κατασκευής FPGA ή από άλλες εταιρίες και μπορεί να διατίθενται είτε δωρεάν είτε επί πληρωμή.

3.4 Οικογένεια FPGA Xilinx® Spartan™ -3E

3.4.1 Εισαγωγή

Η οικογένεια Spartan®-3E είναι ειδικά σχεδιασμένη για να ανταποκρίνεται στις ανάγκες σχεδίασης ηλεκτρονικών κυκλωμάτων υψηλής κλίμακας, όγκου και χαμηλού κόστους. Η πενταμελής οικογένεια προσφέρει πυκνότητες που εκτείνονται από 100,000 σε 1,6 εκατομμύρια πύλες. Βασιζόμενη στην προηγούμενη οικογένεια Spartan-3, η 3E έχει αυξήσει τον όγκο των λογικών μπλοκ ανά μονάδα εισόδου/εξόδου, μειώνοντας σημαντικά το κόστος ανά λογικό μπλοκ. Τα νέα χαρακτηριστικά αυξάνουν την απόδοση και μειώνουν το κόστος ανά μονάδα.

Λόγω του χαμηλού κόστους ανά μονάδα, τα συγκεκριμένα FPGA είναι ιδανικά για μια ευρεία γκάμα ηλεκτρονικών καταναλωτικών προϊόντων, όπως εφαρμογές ευρυζωνικής σύνδεσης, τοπικών δικτύων, εφαρμογών εικόνας και ήχου, ψηφιακής τηλεόρασης κ.α.

Η οικογένεια Spartan-3E είναι μια ανώτερη εναλλακτική λύση έναντι των ολοκληρωμένων κυκλωμάτων ειδικού σκοπού (ASIC), καθώς δεν έχουν υψηλό αρχικό κόστος υλοποίησης, είναι πιο γρήγορα στην υλοποίηση, ελαχιστοποιώντας το χρόνο που χρειάζεται από τη σχεδίαση στη διάθεση του προϊόντος στην αγορά. Επίσης αντίθετα με τα ASICs ο επαναπρογραμματισμός και η αναβάθμιση είναι εύκολη διαδικασία, και δεν χρειάζεται αντικατάσταση του υλικού.

3.4.2 Αρχιτεκτονική Xilinx® Spartan™ -3E

Η Αρχιτεκτονική της οικογένειας Spartan-3E αποτελείται από πέντε θεμελιώδη προγραμματιζόμενα λειτουργικά στοιχεία [15].

- **Διαμορφώσιμα Λογικά Μπλοκ (CLBs)**

Τα διαμορφώσιμα λογικά μπλοκ περιέχουν LUTs (Look-up Tables) βασισμένα σε RAM που υλοποιούν λογικές πράξεις και στοιχεία μνήμης που υλοποιούν flip-flop ή latches. Τα CLBs μπορούν να προγραμματιστούν έτσι ώστε να εκτελούν μια ευρεία ποικιλία λογικών λειτουργιών όπως και λειτουργίες αποθήκευσης δεδομένων.

- **Μπλοκ εισόδου/εξόδου (Input/Output Blocks - IOBs)**

Τα μπλοκ εισόδου/εξόδου ελέγχουν τη ροή δεδομένων μεταξύ των pin εισόδου/εξόδου και της εσωτερικής λογικής. Κάθε μπλοκ υποστηρίζει αμφίδρομη ροή δεδομένων με λειτουργία

τριών καταστάσεων. Υποστηρίζονται αρκετά πρότυπα σήματος, ανάμεσα σε αυτά τέσσερα διαφορεικά υψηλής απόδοσης.

- **RAM μπλοκ**

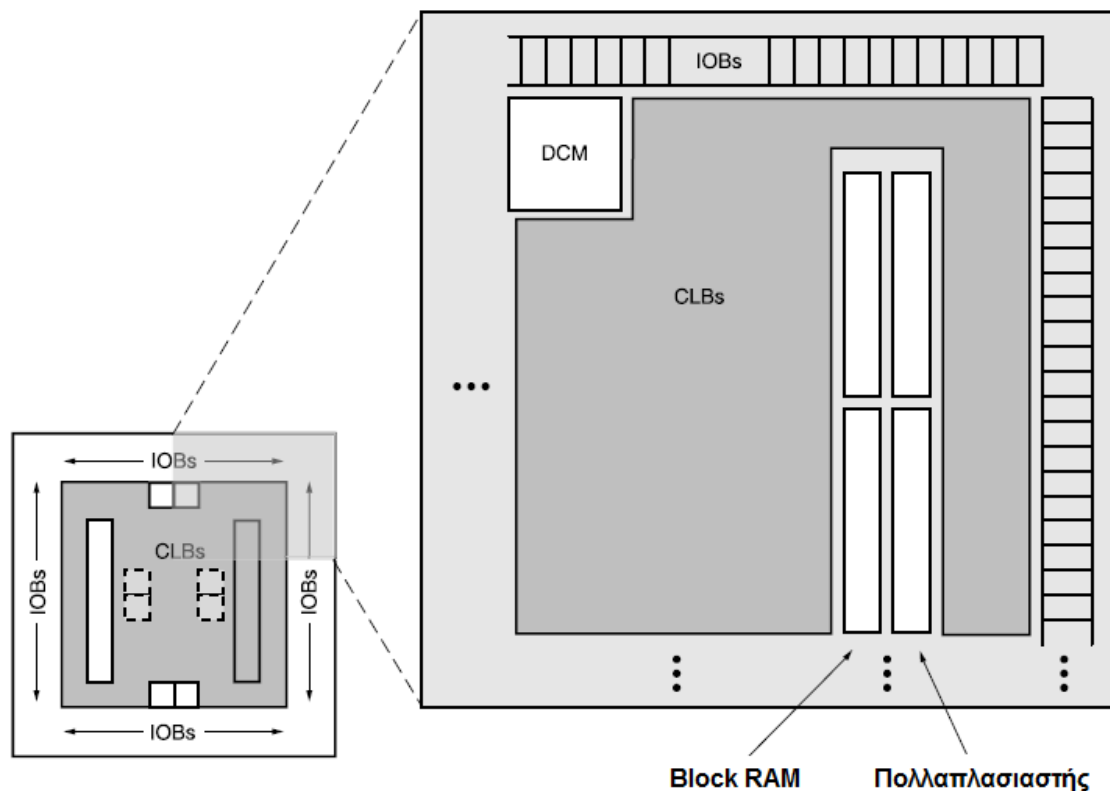
Τα μπλοκ της RAM παρέχουν αποθήκευση δεδομένων σε μορφή διθύρων μπλοκ (dual port) των 18Kbit.

- **Πολλαπλασιαστές**

Οι πολλαπλασιαστές δέχονται σαν εισόδους δύο 18 bit δυαδικούς αριθμούς και υπολογίζουν το γινόμενο.

- **Μπλοκ διαχείρισης ρολογιού (Digital Clock Manager - DCM)**

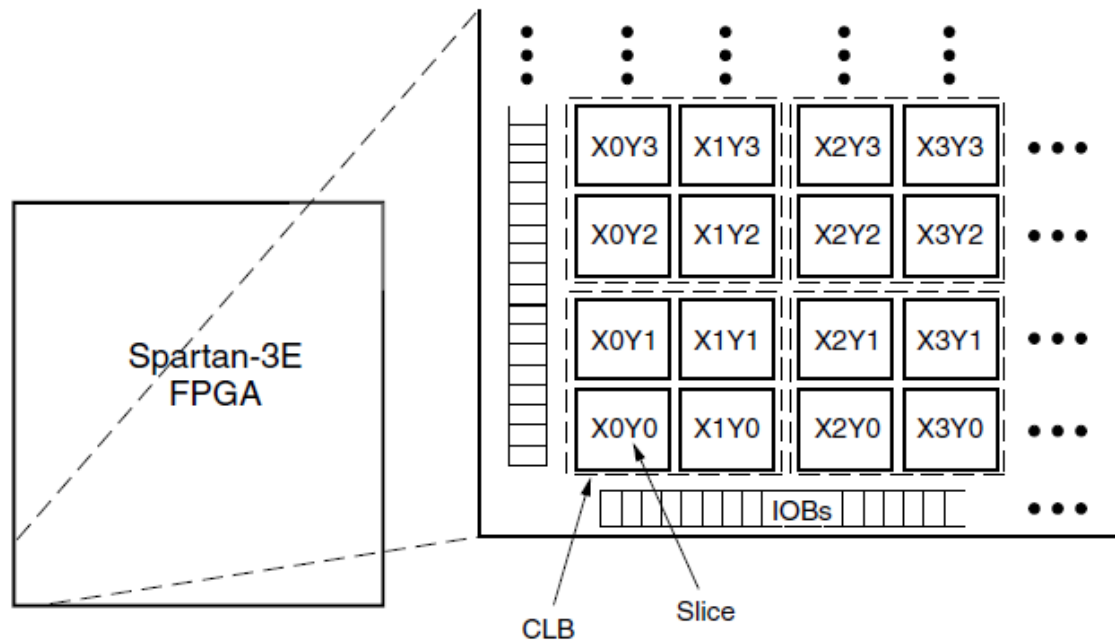
Τα μπλοκ προσφέρουν πλήρως ψηφιακές λύσεις για διανομή, καθυστέρηση, πολλαπλασιασμό, διαίρεση και ολίσθηση φάσης των σημάτων ρολογιού.



Σχήμα 3.4: Αρχιτεκτονική οικογένειας Xilinx Spartan-3E

Περιγραφή των διαμορφώσιμων λογικών μπλοκ (CLBs)

Τα CLB αποτελούν τη κύρια λογική μονάδα για υλοποίηση τόσο συνδυαστικών, όσο και ακολουθιακών κυκλωμάτων. Κάθε CLB περιέχει 4 slices και κάθε slice περιέχει δύο LUT για υλοποίηση συνδυαστικής λογικής και δύο ειδικά στοιχεία αποθήκευσης που μπορούν να χρησιμοποιηθούν σαν flip-flop ή latches. Τα LUTs μπορούν να υλοποιήσουν μία RAM των 16 bit ή ένα καταχωρητή ολίσθησης των 16 bit. Όλα τα CLB είναι πανομοιότυπα και διατάσσονται σε συστοιχίες από γραμμές και στήλες όπως φαίνεται στο Σχ.3.5. Το πλήθος των γραμμών και στηλών των CLB αλλά και το πλήθος της μνήμης και LUT κάθε συσκευής της οικογένειας Spartan-3E παρουσιάζονται στον πίνακα 3.6.



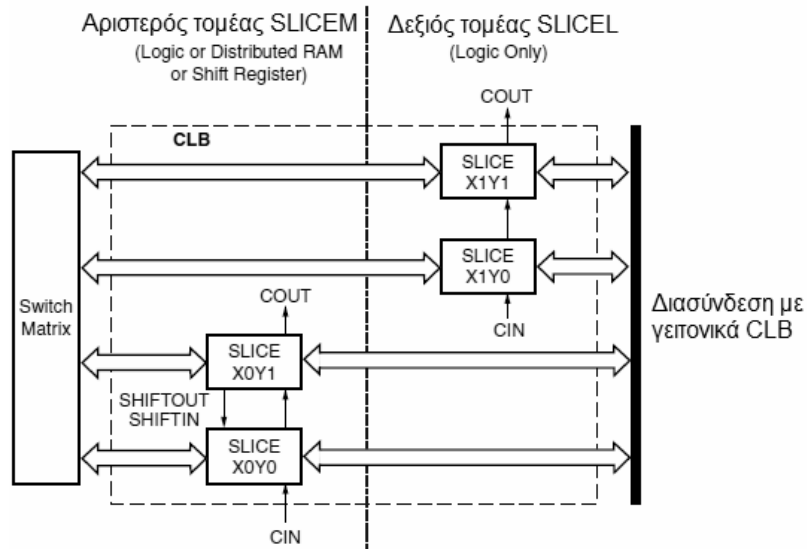
Σχήμα 3.5: CLB και Slices για την αρχιτεκτονική Xilinx Spartan-3E

Device	CLB Rows	CLB Columns	CLB Total ⁽¹⁾	Slices	LUTs / Flip-Flops	Equivalent Logic Cells	RAM16 / SRL16	Distributed RAM Bits
XC3S100E	22	16	240	960	1,920	2,160	960	15,360
XC3S250E	34	26	612	2,448	4,896	5,508	2,448	39,168
XC3S500E	46	34	1,164	4,656	9,312	10,476	4,656	74,496
XC3S1200E	60	46	2,168	8,672	17,344	19,512	8,672	138,752
XC3S1600E	76	58	3,688	14,752	29,504	33,192	14,752	236,032

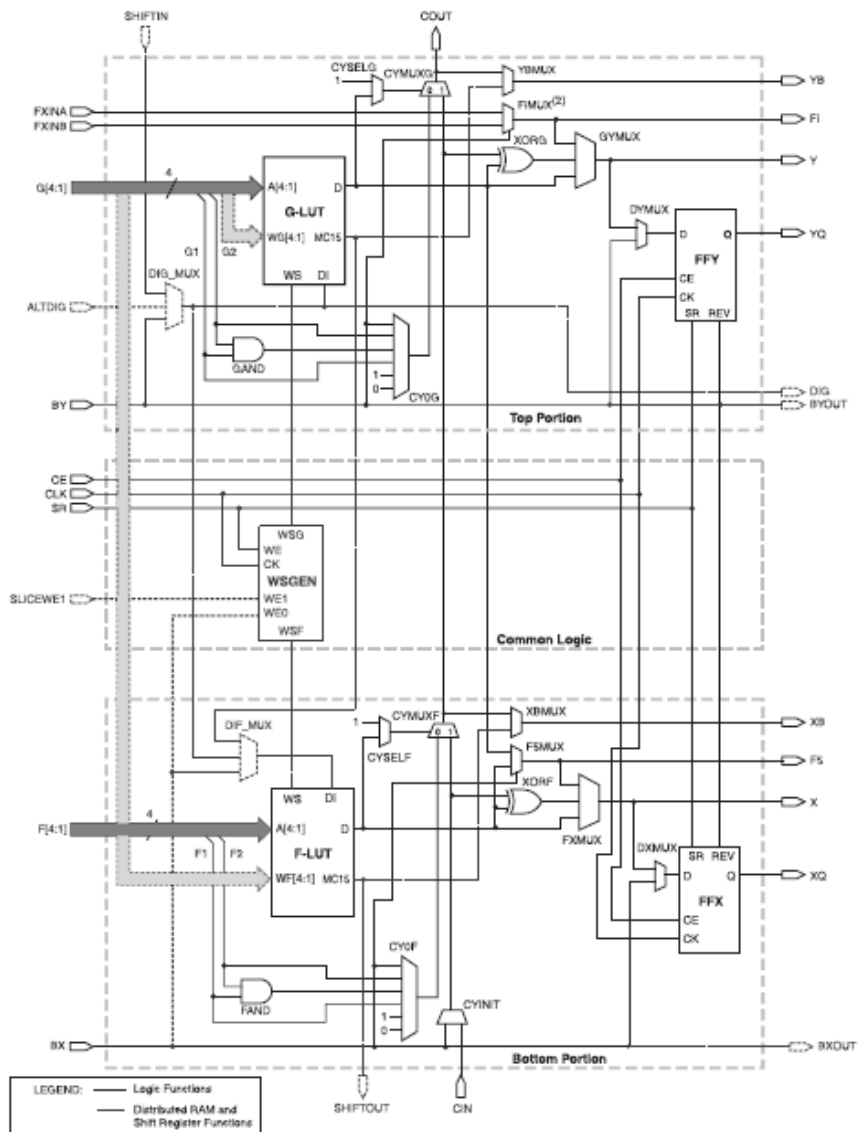
Σχήμα 3.6: Πόροι FPGA οικογένειας Spartan-3E

Μέσα στο Slice:

Κάθε CLB αποτελείται από τέσσερα διασυνδεδεμένα slices Σχ.3.7. Αυτά τα slices κατηγοριοποιούνται σε ζεύγη και κάθε ζεύγος διατάσσεται σε μια στήλη που έχει ξεχωριστά και ανεξάρτητα σήματα κρατουμένου (carry-in, carry-out). Ο αριστερός τομέας υποστηρίζει τόσο λογικές πράξεις, όσο και πράξεις αποθήκευσης και καλείται SLICEM. Ο δεξιός τομέας υποστηρίζει μόνο λογικές πράξεις και καλείται SLICEL. Επομένως οι μισοί LUT μπορούν να υλοποιήσουν λογικές πράξεις και πράξεις αποθήκευσης, ενώ οι άλλοι μισοί μόνο λογικές πράξεις. Τα SLICEL λόγω της μικρότερης πολυπλοκότητάς τους, μειώνουν το μέγεθος και το κόστος κάθε FPGA, ενώ έχουν και καλύτερη απόδοση από τα SLICEM.



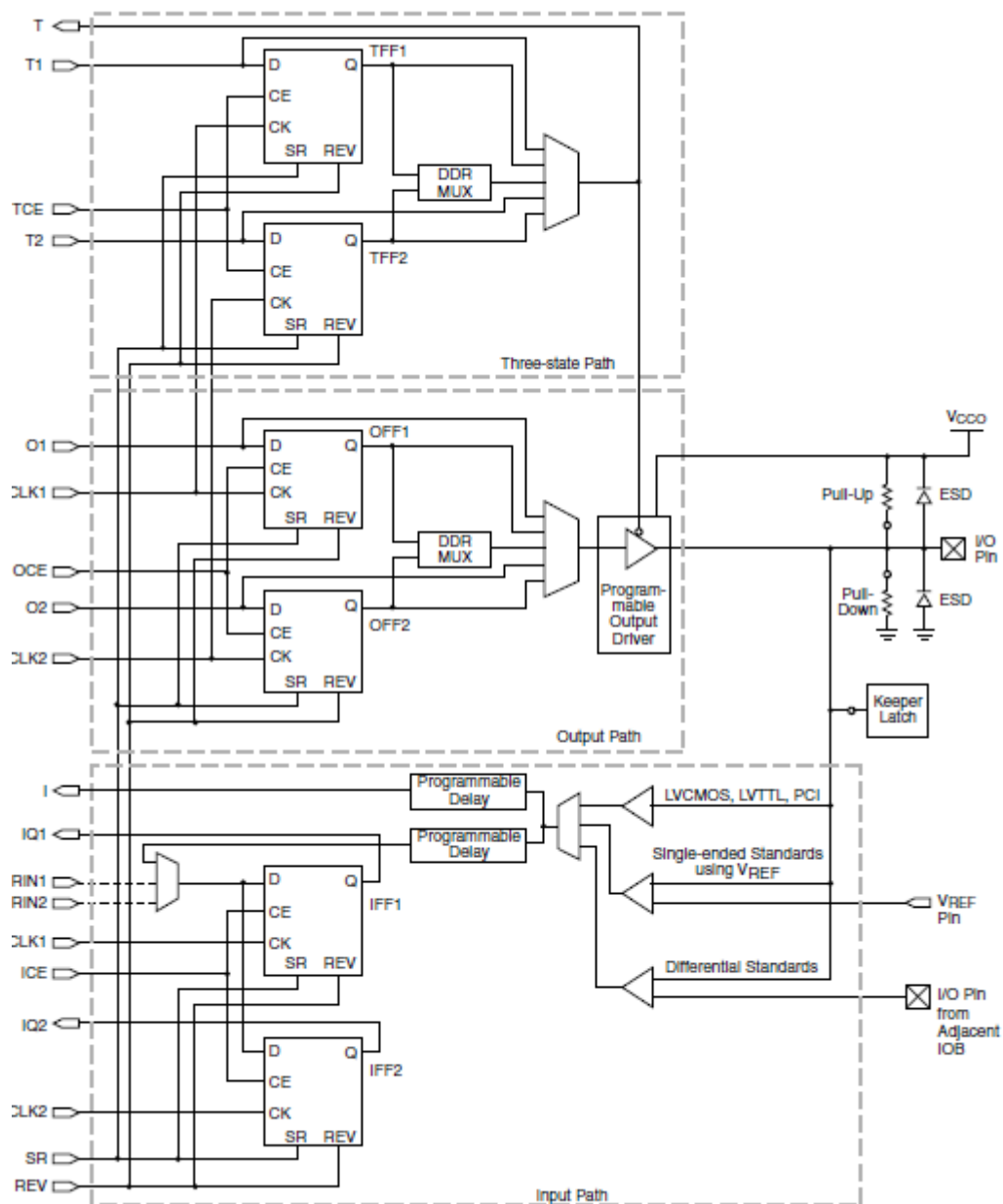
Σχήμα 3.7: Διαρρύθμιση στο εσωτερικό ενός CLB



Σχήμα 3.8: Απλοποιημένο διάγραμμα ενός SLICEM

Περιγραφή των μπλοκ εισόδου-εξόδου (IOBs)

Τα μπλοκ εισόδου-εξόδου προσφέρουν μια προγραμματιζόμενη μονόδρομη ή αμφίδρομη διεπιφάνεια μεταξύ των pin εισόδου-εξόδου και της εσωτερικής λογικής του FPGA. Η μονόδρομη επικοινωνία εκτελείται με μπλοκ εισόδου (input-only) και σε κάθε αρχιτεκτονική το πλήθος τους δεν ξεπερνά το 25% των συνολικών IOBs. Στο Σχ.3.9 παρουσιάζεται μια απλοποιημένη δομή ενός αμφίδρομου IOB. Υπάρχουν τρεις κύριοι δίοδοι σημάτων σε ένα IOB: η δίοδος εισόδου, η δίοδος εξόδου και η δίοδος τριών καταστάσεων (3-state). Κάθε δίοδος έχει το δικό της ζεύγος μονάδων αποθήκευσης που μπορούν να χρησιμοποιηθούν σαν μανδαλωτές ή καταχωρητές. Στη δίοδο εισόδου παρεμβάλλονται στοιχεία προγραμματιζόμενης καθυστέρησης, έτσι ώστε να συγχρονίζονται τα σήματα εισόδου με τα φλιπ-φλοπ IFF1 και IFF2.

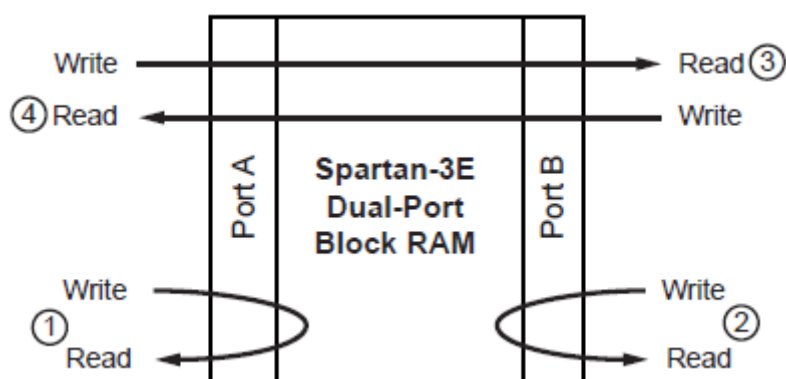


Σχήμα 3.9: Απλοποιημένο διάγραμμα ενός IOB

Περιγραφή των RAM μπλοκ

Η οικογένεια Spartan-3E παρέχει 4 έως 36 RAM μπλοκ που είναι οργανωμένα σε διαμορφώσιμα δίθυρα (dual-port) μπλοκ των 18 Kbits. Τα RAM μπλοκ είναι ιδανικά για σύγχρονη αποθήκευση μεγάλου όγκου δεδομένων ενώ η κατανεμημένη RAM είναι ιδανική για αποθήκευση μικρού όγκου δεδομένων.

Κάθε RAM μπλοκ έχει δομή δύο θυρών (dual-port). Οι δύο πανομοιότυπες θύρες, η θύρα A και η θύρα B που επιτρέπουν την ταυτόχρονη και ανεξάρτητη πρόσβαση στο ίδιο RAM μπλοκ. Κάθε RAM μπλοκ έχει μέγιστη χωρητικότητα 18,432 bit ή 16,384 bit χωρίς τα bit ισοτιμίας.



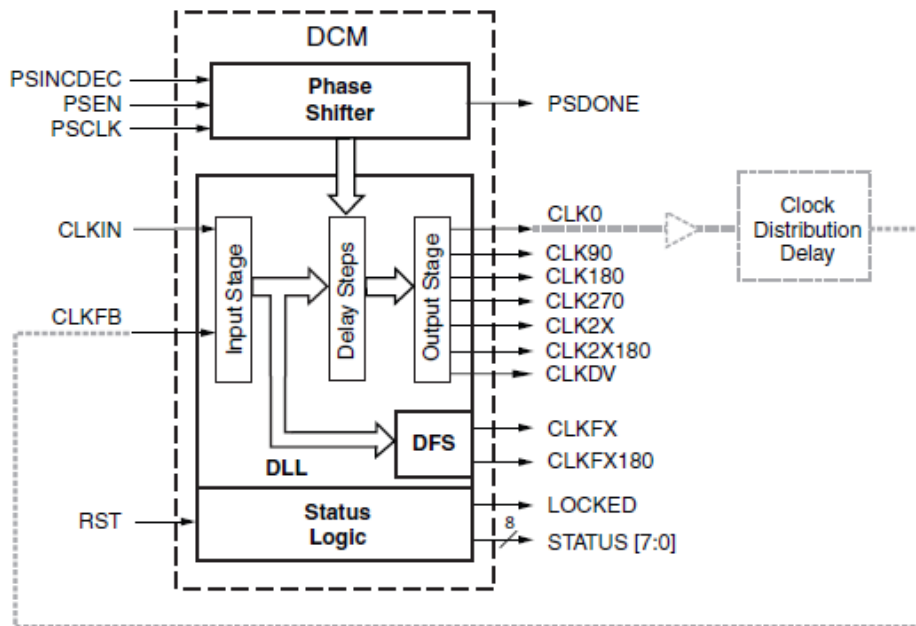
Σχήμα 3.10: Δομή RAM μπλοκ

Τα δεδομένα μπορούν να ακολουθήσουν τέσσερις διαφορετικές διαδρομές κατά την εγγραφή και ανάγνωση από ένα RAM μπλοκ όπως φαίνεται στο Σχ.3.10:

1. Ανάγνωση και εγγραφή από τη θύρα A
2. Ανάγνωση και εγγραφή από τη θύρα B
3. Εγγραφή από τη θύρα A και ανάγνωση από τη θύρα B
4. Εγγραφή από τη θύρα B και ανάγνωση από τη θύρα A

Περιγραφή του μπλοκ διαχείρισης ρολογιού (DCM)

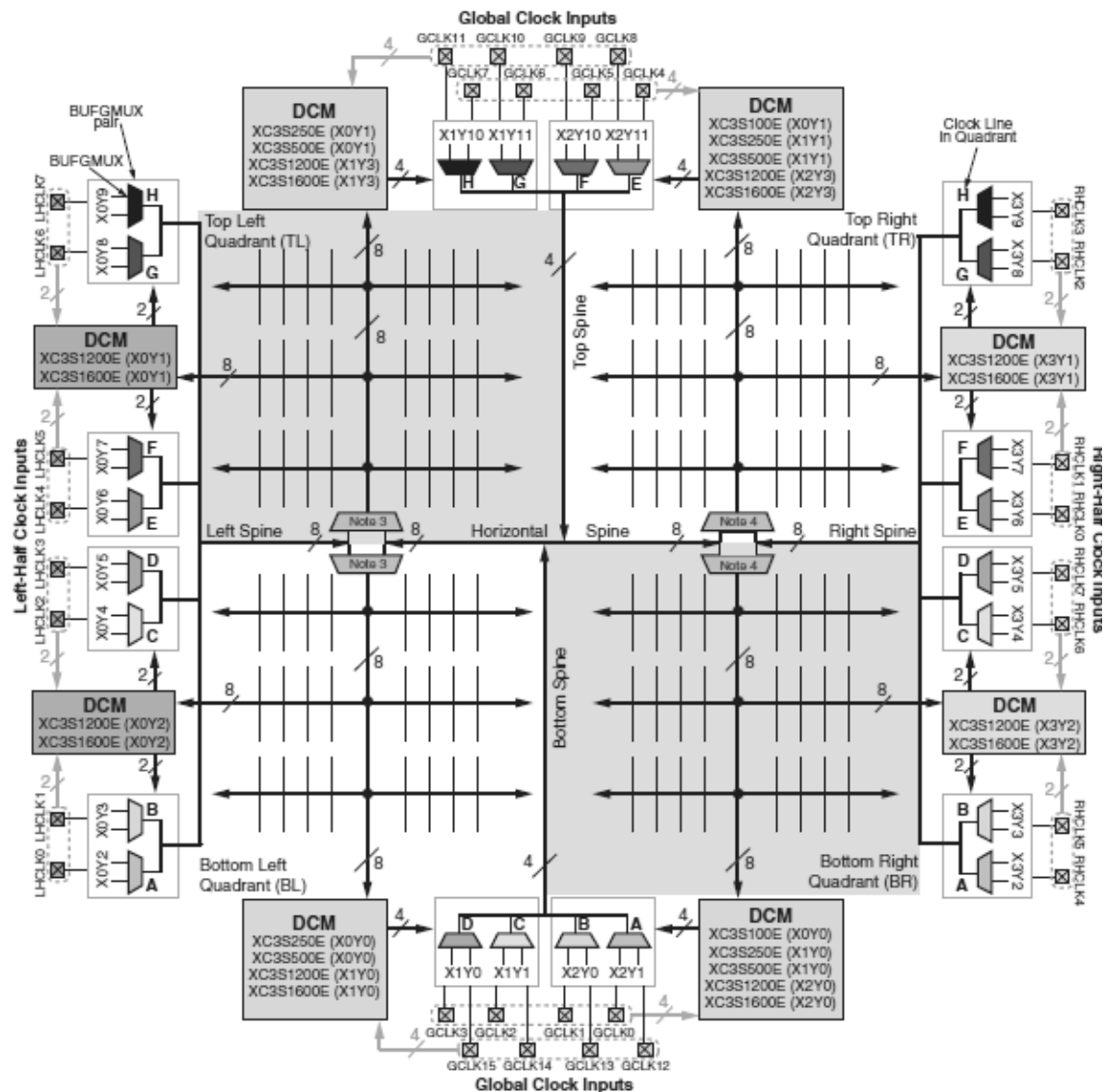
Τα μπλοκ διαχείρισης ρολογιού προσφέρουν πλήρη έλεγχο στη συχνότητα, στην ολίσθηση φάσης και στο skew στα σήματα ρολογιού. Η κύρια μονάδα για επίτευξη αυτού του σκοπού είναι ο βρόχος καθυστέρησης (Delay-Locked Loop, DLL), μια πλήρως ψηφιακή μονάδα που χρησιμοποιεί ανάδραση για να διατηρήσει τα χαρακτηριστικά του σήματος του ρολογιού παρά τις μεταβολές στη θερμοκρασία λειτουργίας και τροφοδοσίας της μονάδας. Μέσω του DCM μπορούν να παραχθούν σήματα διάφορων συχνοτήτων στην έξοδο, πολλαπλασιάζοντας και διαιρώντας το σήμα εισόδου με διάφορους παράγοντες.



Σχήμα 3.11: Δομή DCM μπλοκ

Δίκτυο ρολογιού (Global Clock Network)

Το δίκτυο ρολογιού προσφέρει μια είναι χαμηλής χωρητικότητας, χαμηλού skew διασύνδεση για μεταφορά σημάτων υψηλής συχνότητας στο εσωτερικό του FPGA. Στο δίκτυο περιέχονται και εισοδοί για σήματα ρολογιού (clock inputs) όπως απομονωτές και πολυπλέκτες για οδήγηση σημάτων ρολογιού (BUFG και BUFGMUX). Πρέπει να τονισθεί ότι όλα τα σήματα εισόδου που είναι σήματα ρολογιού, πρέπει να οδηγούνται από το δίκτυο ρολογιού και όχι το δίκτυο διασύνδεσης για αποφυγή καθυστερήσεων και skew.



Σχήμα 3.12: Δίκτυο ρολογιού οικογένειας Spartan-3E

Δίκτυο διασύνδεσης (Interconnecting Network)

Το δίκτυο διασύνδεσης μεταφέρει σήματα μεταξύ των λειτουργικών στοιχείων του FPGA. Το εργαλείο Place & Route (PAR) της Xilinx κάνει χρήση του δικτύου για να δημιουργήσει το επιθυμητό κύκλωμα. Υπάρχουν τέσσερα είδη διασύνδεσης:

➤ Μακριές Γραμμές:

Κάθε μακριά γραμμή εκτείνεται οριζόντια ή κάθετα και συνδέει κάθε έκτο στοιχείο. Οι μακριές γραμμές είναι ιδανικές για μεταφορά σημάτων υψηλής συχνότητας, λόγω της χαμηλής χωρητικότητάς τους. Αν σε κάποια υλοποίηση χρησιμοποιείται όλο το δίκτυο του μπλοκ διαχείρισης ρολογιού, σήματα ρολογιού μπορούν να δρομολογηθούν μέσω του δικτύου μακριών γραμμών.

- **Γραμμές Hex**

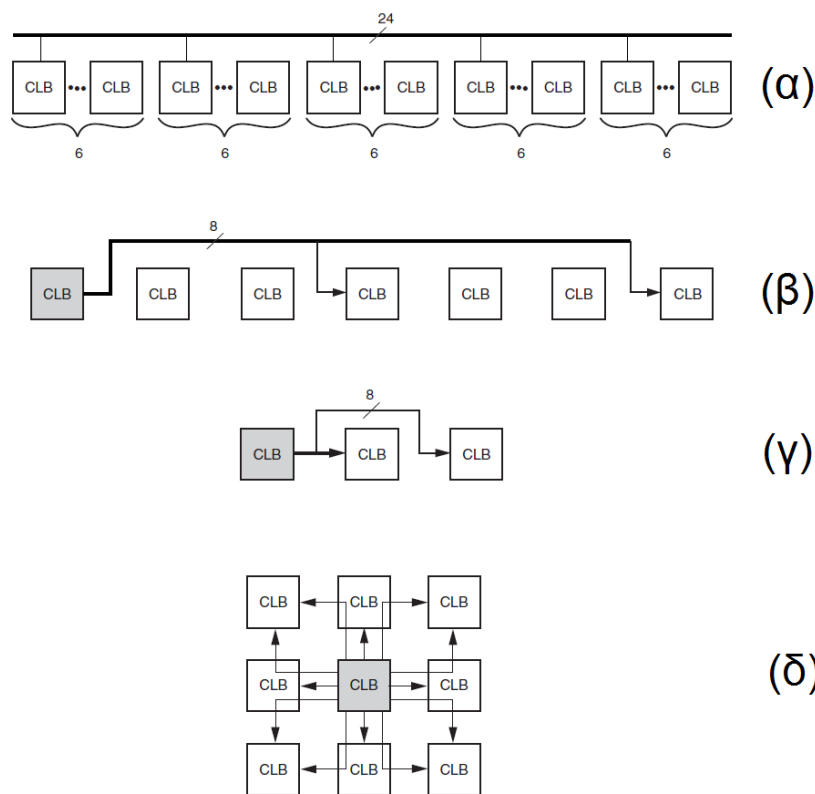
Οι γραμμές Hex συνδέονται με ένα κάθε τρία στοιχεία. Ως προς τα χαρακτηριστικά προσεγγίζουν τις μακριές γραμμές προσφέροντας παράλληλα μεγαλύτερη συνδεσιμότητα.

- **Διπλές Γραμμές**

Συνδέουν κάθε δεύτερο στοιχείο τόσο οριζόντια, όσο και κάθετα. Υπάρχουν 32 διπλές γραμμές μεταξύ οποιονδήποτε δύο στοιχείων, προσφέροντας μεγαλύτερη συνδεσιμότητα και ευελιξία σε σχέση με τις μακριές και hex γραμμές.

- **Απευθείας Γραμμές**

Αυτές οι γραμμές συνδέουν κάθε στοιχείο με τα γειτονικά του. Είναι οι πιο πολύ χρησιμοποιούμενες γραμμές για σύνδεση στοιχείων μεταξύ τους.



Σχήμα 3.13: Δίκτυο διασύνδεσης
 (α) Μακριές γραμμές
 (β) Hex γραμμές
 (γ) Διπλές γραμμές
 (δ) Απευθείας γραμμές

4

Υλοποίηση του πομπού σε τεχνολογία FPGA

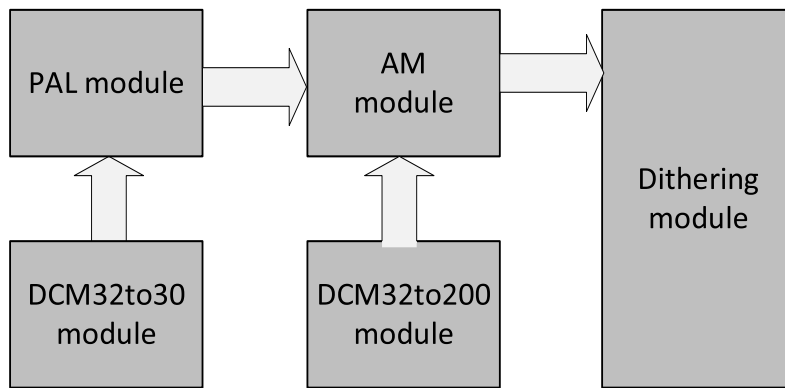
4.1 Περισκόπηση συστήματος

Μια όψη της αρχιτεκτονικής που υλοποιήθηκε στο FPGA εμφανίζεται στο Σχ.4.1. Το σύστημα αποτελείται από πέντε βασικές μονάδες (modules), συγκεκριμένα τη μονάδα κωδικοποίησης εικόνας (PAL module), τη μονάδα διαμόρφωσης πλάτους (AM module), τη μονάδα μετατροπής ψηφιακού σήματος σε αναλογικό (Dithering module) και δύο DCM ip-cores για μετατροπή του ρολογιού εισόδου σε συχνότητες 200 MHz και 20 MHz (DCM32to200 και DCM32to20 modules).

Η μονάδα PAL υλοποιεί μια μηχανή πεπερασμένων καταστάσεων (Finite State Machine, FSM) τύπου Mealy για υλοποίηση του αλγορίθμου κωδικοποίησης αναλογικής τηλεόρασης PAL, παράγει τα σήματα για οδήγηση της ψηφιακής κάμερας και υλοποιεί αλγορίθμους για προσωρινή αποθήκευση των δεδομένων από τη κάμερα και ενσωμάτωσης τους στο σήμα αναλογικής τηλεόρασης. Η μονάδα PAL αποτελείται από διάφορες άλλες υπομονάδες που αναλύονται στο σχετικό υποκεφάλαιο. Το υποσύστημα διαμόρφωσης πλάτους διαμορφώνει ψηφιακά το σήμα από τη μονάδα PAL, μεταφέροντάς το από τη βασική ζώνη στη ζώνη μετάδοσης. Περιέχει το ψηφιακό συνθέτη (Direct Digital Synthesizer, DDS) για παραγωγή του ημιτόνου φέρουσας συχνότητας, όπως και το δυαδικό πολλαπλασιαστή για την επίτευξη AM διαμόρφωσης. Η μονάδα dithering είναι υπεύθυνη για τη μετατροπή του πλήρως ψηφιακού σήματος σε αναλογικό, χωρίς τη χρήση DAC, υλοποιώντας καινοτόμους αλγορίθμους που αναπτυχθήκαν στα εργαστήρια του Τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής του ΕΜΠ [16].

4.2 Υλοποίηση συστήματος σε υλικό

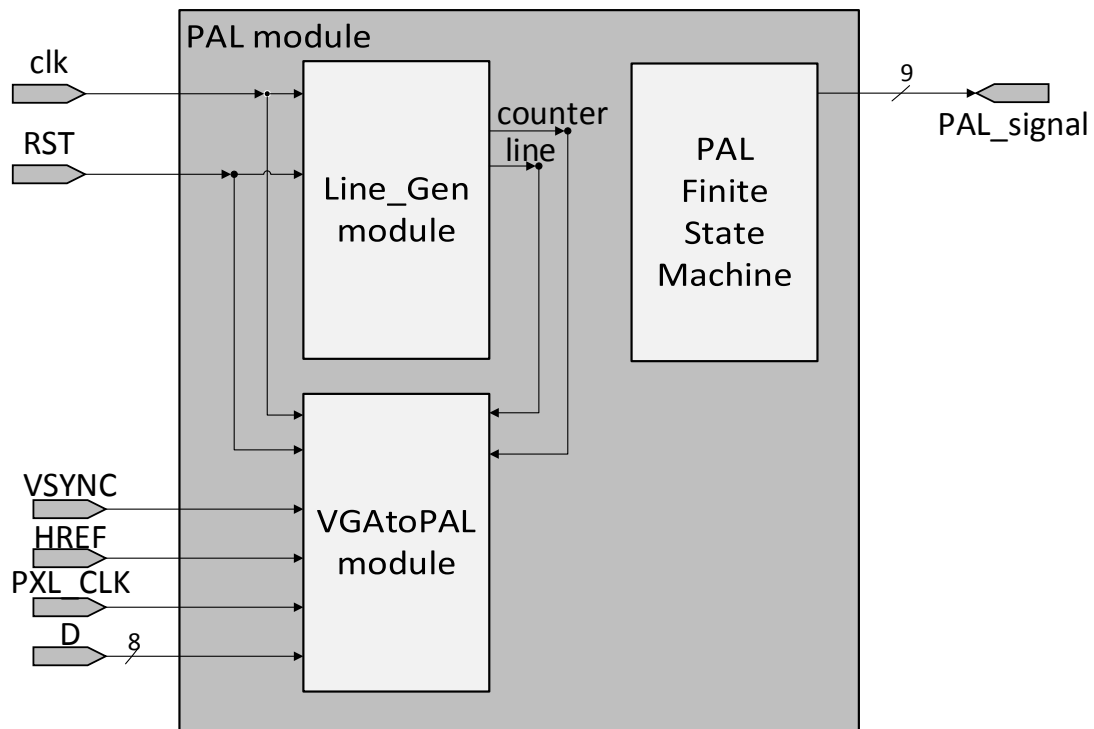
Σε αυτή την ενότητα θα γίνει λεπτομερής περιγραφή των επιμέρους μονάδων που απαρτίζουν την υλοποίηση και τον τρόπο που αυτά συνδέονται μεταξύ τους για την υλοποίηση του πλήρως ψηφιακού πομπού βίντεο σε τεχνολογία FPGA.



Σχήμα 4.1: Αρχιτεκτονική δομή συστήματος

4.2.1 Υπομονάδα PAL (PAL Module)

Η υπομονάδα PAL, όπως υποδεικνύει το όνομά της, είναι υπεύθυνη για την παραγωγή του πλάτους 9 bit κωδικοποιημένου σύνθετου σήματος PAL βασικής ζώνης. Μέσα στην υπομονάδα PAL περιέχονται οι αλγόριθμοι για αποθήκευση μέρους από την εικόνα από τη VGA κάμερα σε RAM προσωρινής αποθήκευσης (frame buffer), για την ανάγνωση από τη RAM και ενσωμάτωσή τους στο παραγόμενο σήμα βασικής ζώνης.



Σχήμα 4.2: Αρχιτεκτονική δομή υπομονάδας PAL

4.2.1.1 Υπομονάδα Line_Gen

Η υπομονάδα Line_Gen παίρνει σαν είσοδο το σήμα ρολογιού clk συχνότητας 20 MHz και το σήμα επαναφοράς (reset) rst και επιστρέφει στην έξοδο τα σήματα counter και line που χρειάζονται για τη λειτουργία του FSM. Για τη λειτουργία του FSM έχει επιλεγεί σήμα ρολογιού συχνότητας 20 MHz. Εφόσον κάθε γραμμή έχει διάρκεια 64μs, χρειάζονται 1280 κύκλοι του ρολογιού των 20 MHz για να ανατρέξει μια γραμμή. Έτσι ο καταχωρητής counter παίρνει τιμές από 0 έως 1279. Ο καταχωρητής line παίρνει τιμές όσες από 0 έως και 624, όσες δηλαδή οι γραμμές του συστήματος PAL. Όταν η είσοδος rst πάρει τιμή 1, γίνεται σύγχρονη αρχικοποίηση των καταχωρητών στη τιμή 0. Ο κώδικας της υπομονάδας παρατίθεται παρακάτω:

Κώδικας υπομονάδας Line_Gen

```
module Line_Generator(input wire clk,rst,
                      output reg [10:0] counter,
                      output reg [9:0] line);

always@(posedge clk)
  if (rst)
    begin
      counter<=11'd0;
      line<=10'd0;
    end
  else if (counter == 1279)
    begin
      counter <= 0;
      if (line == 624 ) line<=0; else line<=line+10'd1;
    end
  else counter<=counter+11'd1;

endmodule
```

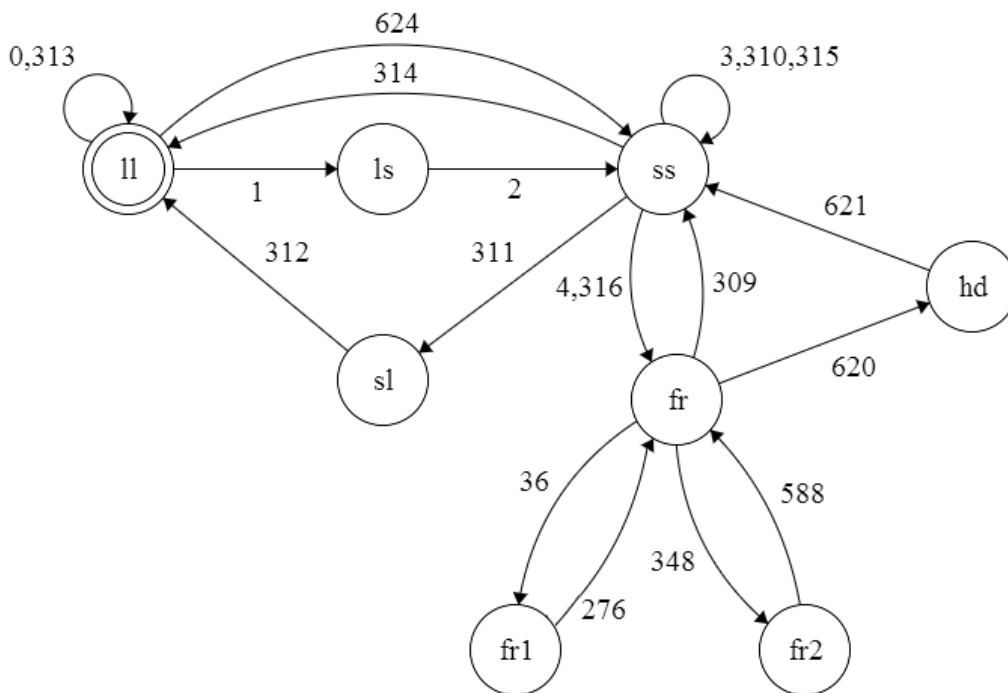
4.2.1.2 Μηχανή Πεπερασμένων Καταστάσεων PAL

Για την παραγωγή των σημάτων κωδικοποίησης αναλογικού σήματος εικόνας PAL σχεδιάστηκε και υλοποιήθηκε μια μηχανή πεπερασμένων καταστάσεων τύπου Mealy Σχ.4.4. Οι καταστάσεις στις οποίες κωδικοποιήθηκε η ΜΠΚ είναι οκτώ και χαρακτηρίζουν κάθε είδος διαφορετικής κατάσταση που θέλαμε να παραγάγουμε. Κάθε κατάσταση διαρκεί 64μs και κωδικοποιεί ένα είδος γραμμής στη κωδικοποίηση PAL. Έτσι η κατάσταση II (συντομογραφία για long_long) αντιστοιχεί σε δύο μεγάλους παλμούς συγχρονισμού, ενώ η fr (συντομογραφία για frame) κωδικοποιεί ένα μαύρο πλαίσιο γύρω από την εικόνα που προβάλλεται. Στον πίνακα 4.3 παρουσιάζονται οι καταστάσεις που κωδικοποιήθηκαν και οι γραμμές στις ποιες αντιστοιχούν.

Κατάσταση	Συντομογραφία	Επεξήγηση	Γραμμή στην οποία εμφανίζονται
long_long	ll	Δύο μακριοί παλμοί συγχρονισμού	0,1,313,314
long_short	ls	Ένας μακρύς και ένας βραχύς παλμός συγχρονισμού	2
short_short	ss	Δύο βραχύς παλμοί συγχρονισμού	3,4,310,311,622,623,624
short_long	sl	Ένας κοντός και ένας μακρύς παλμός συγχρονισμού	312
half_data	hd	Μισή γραμμή εικόνας	621
frame	fr	Μαύρο πλαίσιο γύρω από εικόνα	5-37,277-310, 317-349,589-621
data_field_1	df1	Γραμμές Πλαισίου 1	38-276
data_field_2	df2	Γραμμές Πλαισίου 2	349-589

Πίνακας 4.3: Κωδικοποίηση μηχανής πεπερασμένων καταστάσεων

Με βάση τον παραπάνω πίνακα, με τις καταστάσεις που αναφέρονται παραπάνω και με εισόδους τη παρούσα τιμή του μετρητή line, σχεδιάστηκε η παρακάτω μηχανή πεπερασμένων καταστάσεων.



Σχήμα 4.4: Διάγραμμα μηχανής πεπερασμένων καταστάσεων PAL

Ο κώδικας χωρίζεται σε δύο μέρη, στη λογική της επόμενης κατάστασης (next state logic) και τη λογική της εξόδου (output logic). Στη λογική της επόμενης κατάστασης, με βάση την παρούσα κατάσταση και την παρούσα είσοδο επιλέγεται η επόμενη κατάσταση, ενώ στη λογική της εξόδου, με βάση την παρούσα κατάσταση, οδηγούνται διάφορα σήματα στην έξοδο του FSM. Παρακάτω παρουσιάζεται η λογική της επόμενης κατάστασης.

Κώδικας λογικής επόμενης κατάστασης

```
// Next State Logic
always @(posedge clk)
  if (rst) state <= long_long;
  else
    begin
      case (state)
        long_long : if (line == 2) state <= long_short;
                   else if (line == 315) state <= short_short;

        long_short : if (line == 3) state <= short_short;

        short_short: if (line == 5 || line == 317) state <= frame;
                   else if (line == 0) state <= long_long;
                   else if (line == 312) state <= short_long;

        short_long  : if (line == 313) state <= long_long;

        half_data   : if (line == 622) state <= short_short;

        data_field_1: if (line == 277) state <= frame;

        data_field_2: if (line == 589) state <= frame;

        frame       : if (line == 37) state <= data_field_1;
                   else if (line == 310) state <= short_short;
                   else if (line == 349) state <= data_field_2;
                   else if (line == 621) state <= half_data;

      endcase
    end
end
```

Σε κάθε θετική ακμή του ρολογιού γίνεται έλεγχος σε ποια κατάσταση βρίσκεται το FSM. Ανάλογα με τη τρέχουσα γραμμή, γίνεται η εναλλαγή στην επόμενη κατάσταση με βάση το διάγραμμα του Σχ.4.4. Τέλος με πίεση του κουμπιού επαναφοράς (rst) η ακολουθία ξεκινά από την πρώτη κατάσταση long_long.

Στη λογική εξόδου χρησιμοποιούνται τόσο οι καταστάσεις, όσο και ένας μετρητής που δείχνει σε ποιο σημείο ή pixel της γραμμής βρίσκεται το FSM. Ανάλογα με τη τιμή του μετρητή επιλέγεται η έξοδος, με 1 να είναι το επίπεδο συγχρονισμού, το 153 να είναι το επίπεδο του απόλυτου μαύρου και το 511 να αντιστοιχεί στο απόλυτο λευκό.

Κώδικας λογικής εξόδου

```
//Output Logic
always @ (posedge clk)
  case(state)
    long_long : if (counter == 0) PAL_signal <= 1;
               else if (counter == 545) PAL_signal <= 153;
               else if (counter == 639) PAL_signal <= 1;
               else if (counter == 1185) PAL_signal <= 153;

    long_short: if (counter == 0) PAL_signal <= 1;
```

```

else if (counter == 545) PAL_signal <= 153;
else if (counter == 639) PAL_signal <= 1;
else if (counter == 686) PAL_signal <= 153;

short_long : if (counter == 0) PAL_signal <= 1;
else if (counter == 46) PAL_signal <= 153;
else if (counter == 639) PAL_signal <= 1;
else if (counter == 1185) PAL_signal <= 153;

short_short :if (counter == 0) PAL_signal <= 1;
else if (counter == 46) PAL_signal <= 153;
else if (counter == 639) PAL_signal <= 1;
else if (counter == 686) PAL_signal <= 153;

half_data : if (counter == 0) PAL_signal <= 153;
else if (counter == 29) PAL_signal <= 1;
else if (counter == 123) PAL_signal <= 153;

//Chrominance Burst Start
else if (counter == 141) PAL_signal <= 153;
//Chrominance Burst End

else if (counter == 186) PAL_signal <= 153;

//Half Data
else if (counter == 239) PAL_signal <= 153;
else if (counter == 369) PAL_signal <= 204;
else if (counter == 449) PAL_signal <= 255;
else if (counter == 629) PAL_signal <= 306;

//Sync Pulse
else if (counter == 639) PAL_signal <= 1;
else if (counter == 686) PAL_signal <= 153;

data_field_1:if (counter == 0) PAL_signal <= 153;
else if (counter == 29) PAL_signal <= 1;
else if (counter == 123) PAL_signal <= 153;

//Chrominance Burst Start
else if (counter == 141) PAL_signal <= 153;
//Chrominance Burst End

else if (counter == 186) PAL_signal <= 153;

//PAL Visible Signal
else if (counter == 239) PAL_signal <= 153;
else if (counter >= 359 & counter <1159)
PAL_signal <= PAL_SIG;
else if (counter == 1159) PAL_signal <= 153;
//800 pixel

data_field_2 : if (counter == 0) PAL_signal <= 153;
else if (counter == 29) PAL_signal <= 1;
else if (counter == 123) PAL signal <= 153;

```

```

//Chrominance Burst Start
else if (counter == 141) PAL_signal <= 153;
//Chrominance Burst End

else if (counter == 186) PAL_signal <= 153;

//PAL Visible Signal
else if (counter == 239) PAL_signal <= 153;
else if (counter >= 359 & counter <1159)
PAL_signal <= PAL_SIG;
else if (counter == 1159) PAL_signal <= 153;
//800 pixel

frame :   if (counter == 0) PAL_signal <= 153;
           else if (counter == 29) PAL_signal <= 1;
           else if (counter == 123) PAL_signal <= 153;

           //Chrominance Burst Start
           else if (counter == 141) PAL_signal <= 153;
           //Chrominance Burst End

           else if (counter == 186) PAL_signal <= 153;

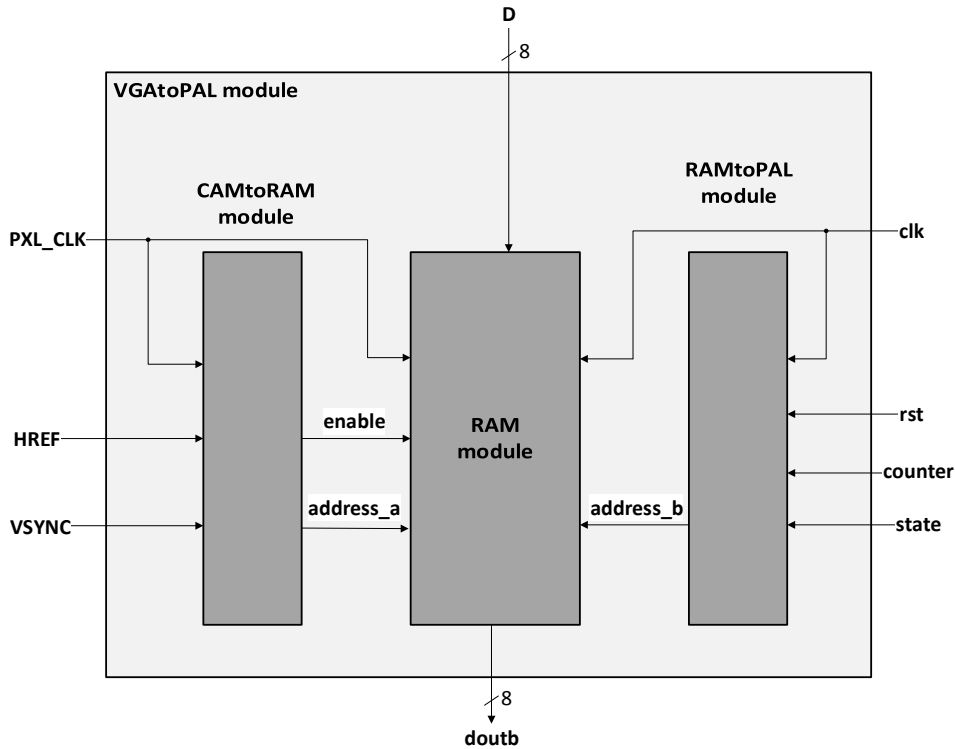
           //Black Frame
           else if (counter == 239) PAL_signal <= 153;

endcase

```

4.2.1.3 Υπομονάδα VGAtοPAL (VGAtοPAL Module)

Στην υπομονάδα VGAtοPAL, γίνεται η ανάγνωση μέρους της ασπρόμαυρης εικόνας από την κάμερα, η προσωρινή αποθήκευσή του σε RAM και ακολούθως η τροφοδότησή του, την κατάλληλη στιγμή στο PAL FSM.



Σχήμα 4.5: Αρχιτεκτονική δομή υπομονάδας VGAtPAL

4.2.2.3.a Υπομονάδα CAMtoRAM

Σε αυτή την υπομονάδα γίνεται η ανάγνωση από τη κάμερα, διευθυνσιοδότηση στη RAM και ακολούθως εγγραφή σε αυτή. Η υπομονάδα παίρνει σαν είσοδο το σήμα ρολογιού από τη κάμερα (PXL_CLK) και τα σήματα οριζόντιου και κάθετου συγχρονισμού και (HREF και VSYNC). Χρησιμοποιούνται δύο μετρητές, οι pixel και VGA_line. Σε κάθε γραμμή μεταδίδονται $2 \times 640 = 1280$ byte, ενώ υπάρχουν 480 γραμμές. Στο πρώτο μέρος υλοποιείται ο μετρητής pixel που μετρά από το 0 έως και το 1279 και ο μετρητής VGA_line που μετρά από το 0 έως και το 479. Στη συνέχεια υλοποιείται ο μετρητής διευθυνσιοδότησης της RAM, address_a. Ο μετρητής address_a αυξάνεται κατά ένα μόνο εφόσον η εικόνα βρίσκεται στη 1^η από 4 γραμμές και επιλέγει το 4^ο από κάθε 8 byte. Έτσι στη RAM γράφεται ένα μέρος της εικόνας, μεγέθους 160x120 εικονιστοιχειών Σχ.4.6.

Κώδικας υπομονάδας CAMtoRAM

```

module CAMtoRAM(input wire PXL_CLK, HREF, VSYNC,
                output reg [14:0] address_a,
                output reg enable);

reg [10:0] pixel=0;
reg [8:0] VGA_line=0;

always @(posedge PXL_CLK)
  if (VSYNC)
  begin
    pixel <= 10'd0;
    VGA_line <= 9'd0;
  end
  else if (HREF)

```

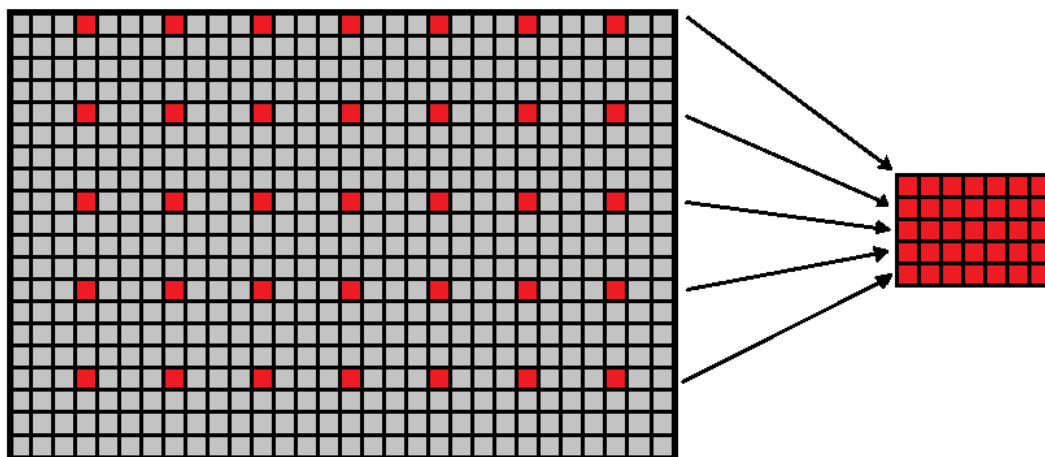
```

if (pixel == 1279)
begin
  pixel <= 0;
  if (VGA_line==479) VGA_line <= 0;
  else VGA_line <= VGA_line +1;
end
else pixel <= pixel +1;

always @(posedge PXL_CLK)
if (VSYNC) address_a <= 0;
else if (HREF & VGA_line[1:0] == 0)
begin
  enable<=(pixel[2:0]==3'd4);
  if (enable)
  if (address_a == 19199) address_a <= 15'd0;
  else address_a <= address_a +15'd1;
end

endmodule

```



Σχήμα 4.6: Μέθοδος υποδειγματοληψίας εικόνας

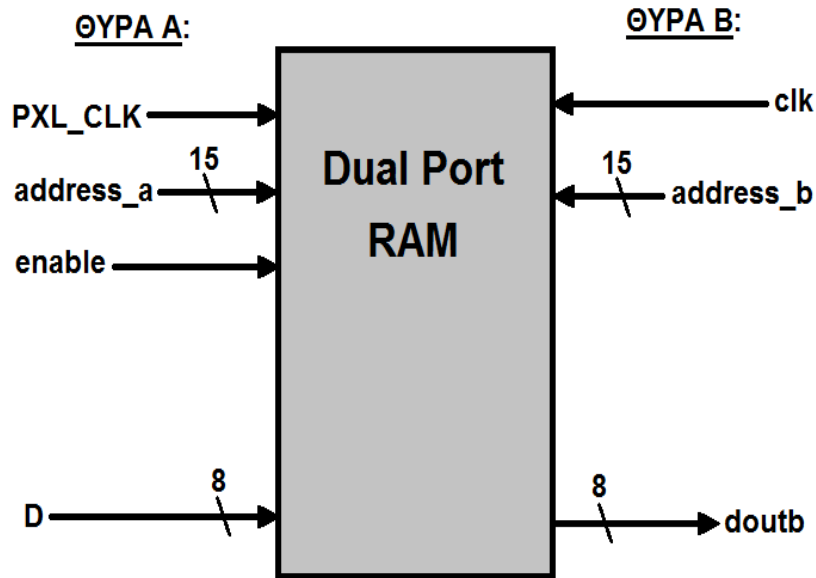
4.2.2.3.β Υπομονάδα RAM

Η υπομονάδα RAM [17] χρησιμοποιείται για την προσωρινή αποθήκευση των εικονιστοιχείων φωτεινότητας από τη κάμερα, για την αργότερα χρησιμοποίησή τους στο FSM του σήματος PAL. Το FPGA Spartan 3EXC3S250E διαθέτει 216 Kbits RAM, σε μορφή μπλοκ των 18 Kbits. Με βάση τον περιορισμό στη RAM, επιλέχθηκε η αποθήκευση εικόνας μεγέθους 160x120 εικονοστοιχείων και βάθους χρώματος 8 bit. Ο συνολικός χώρος που καταλαμβάνεται στη Block RAM είναι:

$$160 \times 120 \times 8 \text{ bit} = 153600 \text{ bits} = 150 \text{ Kbits}$$

Για την υλοποίηση της RAM μεγέθους 150 Kbits χρησιμοποιήθηκε ο *IP CORE Generator and Architecture Wizard* που προσφέρεται στο αναπτυξιακό πακέτο της Xilinx. Η μνήμη πρέπει να γράφει από τη κάμερα και να διαβάζει ταυτόχρονα για το FSM. Για αυτό το λόγο

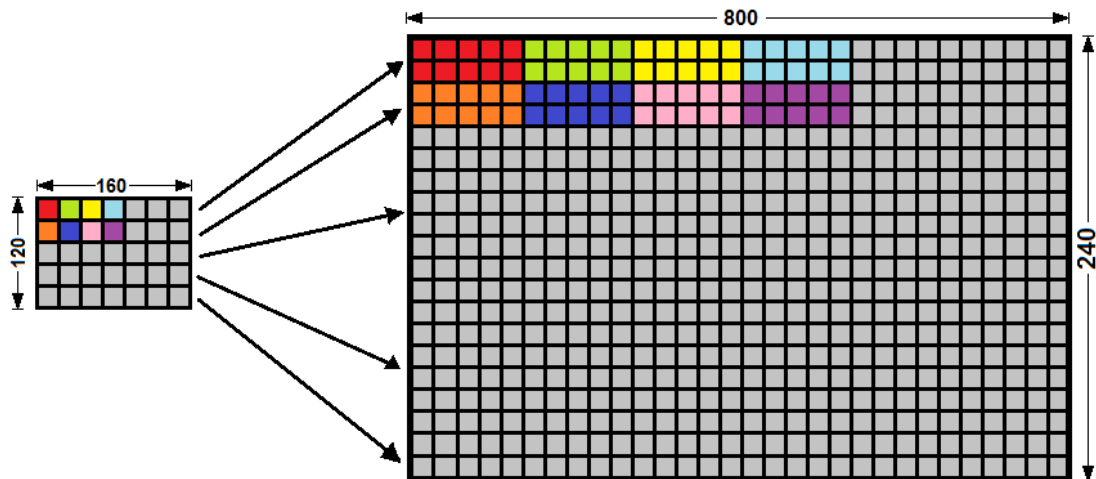
επιλέγηκε η υλοποίηση απλής δύο θυρών RAM (Simple Dual Port RAM). Στη θύρα A μπαίνουν τα σήματα προερχόμενα από την κάμερα και την υπομονάδα VGAtοRAM ενώ στη θύρα B μπαίνουν τα σήματα από την υπομονάδα RAMtoPAL και φεύγουν τα δεδομένα από το δίαυλο doutb για την υπομονάδα PAL.



Σχήμα 4.7: Δομή απλής δύο θυρών RAM (simple dual-port RAM)

4.2.2.3.γ Υπομονάδα RAMtoPAL

Σε αυτή την υπομονάδα υλοποιούνται οι μετρητές, έτσι ώστε τα δεδομένα από τη RAM, να είναι διαθέσιμα την κατάλληλη χρονική στιγμή για ενσωμάτωση στις γραμμές εικόνας που μεταδίδονται. Για αυτό το σκοπό χρειάζονται τα σήματα clk, rst, counter και state από το FSM και μετρητές που υλοποιούνται εντός της υπομονάδας. Οι μετρητές είναι οι cam_pixel και addr_pixel για τα εικονοστοιχεία κάθε γραμμής και cam_line για τις γραμμές που είναι αποθηκευμένες στη RAM και ένας καταχωρητής για επίτρεψη pxl_enable. Ο μετρητής cam_pixel μετρά από το 0 έως και το 799 ενώ ο μετρητής cam_line μετρά από το 0 έως και το 479 για να σχηματίσουν μια εικόνα διαστάσεων 800x240. Αυτή η εικόνα επαναλαμβάνεται και στα δύο πλαίσια της πεπλεγμένης εικόνας που μεταδίδεται. Για να σχηματιστεί η εικόνα, κάθε εικονοστοιχείο επαναλαμβάνεται 5 φορές, ενώ κάθε γραμμή επαναλαμβάνεται 2 φορές.



Σχήμα 4.8: Μέθοδος μεγέθυνσης εικόνας

Παρακάτω παρουσιάζεται ο κώδικας της υπομονάδας:

Κώδικας υπομονάδας RAMtoPAL

```

module RAMtoPAL(input wire clk,rst,
                input wire [10:0] counter,
                input wire [2:0] state,
                output reg [14:0] address_b);

reg [9:0] cam_pixel=0;
reg [8:0] cam_line = 0;
reg pxl_en;
reg [2:0] addr_pixel = 0;

always@(posedge clk)
begin
//Address change enabling for Field 1 or Field 2
// starting from counter position 359
if ((state ==4 || state == 5)&& counter == 359) pxl_en <= 1;
else if ((state ==4 || state == 5)&& counter == 1159)
pxl_en <= 0; //800 pixel horizontal
end

always@(posedge clk )
begin
//Line Counter
if (cam_pixel==799 || rst)
if (cam_line == 479) cam_line <= 0;
else cam_line <= cam_line +1;

//Pixel Counter
if (pxl_en)
begin
if (cam_pixel==799) begin cam_pixel <=0;end
else cam_pixel <= cam_pixel + 1;
end
end

```

```

//For address changing every 5 pixels
if (addr_pixel == 4) addr_pixel <= 0;
else addr_pixel <= addr_pixel +1 ;
end
end

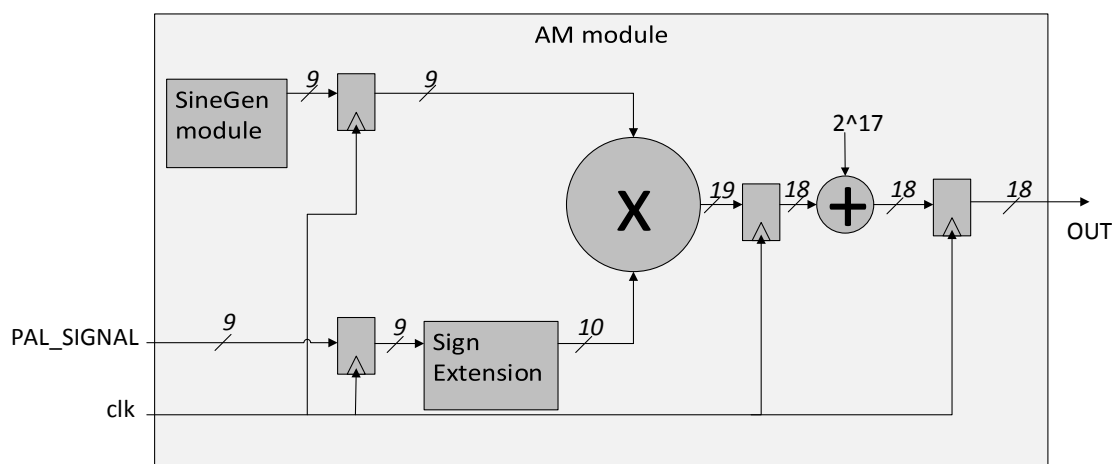
always@(posedge clk)
begin
//Same pixel repeating 5 times
if (rst | state ==7) address_b <= 15'd0;
if (pxl_en && addr_pixel == 4)
//Same line repeating twice
if (cam_pixel==799 && cam_line[0] == 0)
address_b <= address_b - 159;

else if (address_b == 19199) address_b <= 15'd0;
else address_b <= address_b + 1;
end
endmodule

```

4.2.3 Διαμόρφωση Πλάτους (AM)

Στο κεφάλαιο 1 αναλύθηκαν όλες οι μέθοδοι αναλογικής διαμόρφωσης πλάτους. Σε αυτή την παράγραφο περιγράφεται το πως υλοποιήθηκε η διαμόρφωση πλάτους στο FPGA. Όπως φαίνεται στο Σχ. 4.9, υλοποιήθηκε ένας ψηφιακός διαμορφωτής διπλής πλευρικής ζώνης AM-DSB, για εξοικονόμηση πόρων στο FPGA. Με την παρούσα τοπολογία χρησιμοποιείται μόνο ένας από τους 12 διαθέσιμους πολλαπλασιαστές. Αναλυτικά το πλάτους 9-bit κωδικοποιημένο σήμα βασικής ζώνης PAL, πολλαπλασιάζεται με το ημιτονοειδές φέρον πλάτους 9-bit. Στην έξοδο του δυαδικού πολλαπλασιαστή παίρνουμε το AM διαμορφωμένο ψηφιακό σήμα πλάτους 18-bit. Το αποτέλεσμα του πολλαπλασιασμού είναι σε προσημασμένη μορφή και παίρνει τιμές από -2^{17} έως $2^{17}-1$. Για να είναι δυνατή η μετατροπή του σήματος από ψηφιακό σε αναλογικό, πρέπει το σήμα να μετατραπεί σε δυαδική μορφή χωρίς πρόσημο. Αυτό γίνεται προσθέτοντας στο αποτέλεσμα του πολλαπλασιασμού 2^{17} .



Σχήμα 4.9: Δομή υπομονάδας διαμόρφωσης πλάτους

4.2.3.α SineGen module

Για την παραγωγή ημιτονοειδούς κυματομορφής χρησιμοποιήθηκε ο *IP CORE Generator and Architecture Wizard* για δημιουργία ενός Direct Digital Synthesizer [18]-[20]. Οι παράμετροι του DDS παρουσιάζονται στον παρακάτω πίνακα:

Παράμετρος	Τιμή
Configuration	Phase Generator and SIN COS LUT
System Clock (F_{clk})	200 MHz
Phase Width (B)	20
Output Width	9
Phase programmability	Fixed
Phase offset programmability	None
Output Selection	Sine
Channel Phase Angle Increment Values In Binary ($\Delta\theta$)	111101110000101001

Με αυτές τις παραμέτρους υλοποιείται μια γεννήτρια ημιτόνου με ψηφιακή έξοδο πλάτους 9-bit. Η συχνότητά της γεννήτριας ημιτόνου ορίζεται από τη τιμή της ψηφιακής λέξης (phase angle increment) και δίνεται από τον τύπο:

$$f_{out} = \frac{f_{clk} \cdot \Delta\theta}{2^B}$$

Με τις παραπάνω τιμές η συχνότητα του ημιτόνου προγραμματίζεται στα 48.25 MHz, που αντιστοιχεί στη συχνότητα φέροντος βίντεο για το 2^ο κανάλι τηλεόρασης στα VHF. Πριν και μετά τη διαμόρφωση πλάτους που γίνεται με πολλαπλασιασμό, παρεμβάλλονται απομονωτές (buffers) που υλοποιούνται με flip-flop, έτσι ώστε να τηρούνται οι χρονικοί περιορισμοί (timing constraints) του συστήματος. Σε αυτή την υλοποίηση το πλήθος των απομονωτών M επιλέχθηκε να είναι 12.

Παρακάτω παρατίθεται ο κώδικας για υλοποίηση του διαμορφωτή πλάτους.

Κώδικας AM module

```
`timescale 1ns / 1ps

module AM (input wire clk,rst,
           input wire [8:0] PAL SIGNAL,
           output reg [17:0] OUT);

wire signed [8:0] sine;

SineGen SineGen (.clk(clk),.sclr(rst),.sine(sine));

reg signed [8:0] sine_b0;

reg signed [8:0] PAL_Signed_b0;

always @(posedge clk)
begin
sine_b0 <= sine;
end
```

```

    PAL_Signed_b0 <= -{1'b0, PAL_SIGNAL};
end

reg [17:0] pr_0;

reg signed [17:0] OUT;

always @(posedge clk) pr_0 <= sine_b0*PAL_Signed_b0;

always @(posedge clk)
begin
    OUT <= pr_0 + 18'd131072;
end

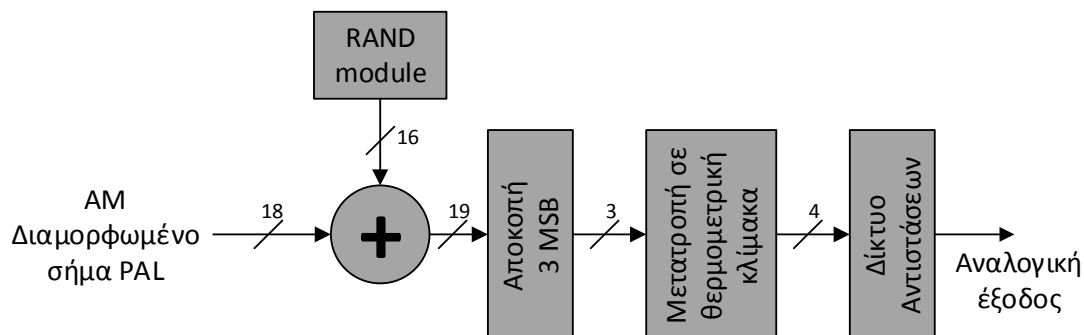
endmodule

```

4.2.4 Μετατροπή ψηφιακού σήματος σε αναλογικό (Random number Dithering)

Υπάρχουν πολλές τεχνικές για μετατροπή του ψηφιακού σήματος σε αναλογικό. Οι πιο γνωστές είναι με χρήση δικτύων αντιστάσεων (binary weighted, δίκτυο R/2R, Θερμομετρικής κλίμακας κ.α) ή με τη μέθοδο της Σ-Δ υπερδειγματοληψίας. Και οι δύο μέθοδοι παρουσιάζουν μειονεκτήματα. Στη μέθοδο του δικτύου αντιστάσεων θα χρειαζόταν να γίνει χρήση πολλών αντιστάσεων, ενώ η μέθοδος της Σ-Δ υπερδειγματοληψίας ενώ είναι καλή στις χαμηλές συχνότητες, για ψηλές συχνότητες χρειάζεται σήμα ρολογιού πολύ υψηλής συχνότητας, που είναι δύσκολο να κατασκευαστεί και είναι πιο δύσκολη μέθοδος στην υλοποίησή της.

Η μέθοδος που χρησιμοποιούμε αίρει και τα δυο μειονεκτήματα που αναφέρθηκαν παραπάνω, καθιστώντας δυνατή τη μετατροπή του ψηφιακού σήματος σε αναλογικό [19],[20], χρησιμοποιώντας όσα λιγότερα αναλογικά εξαρτήματα γίνεται και κάνοντας χρήση των υπάρχουσών τεχνολογιών. Στο Σχ.4.10 παρουσιάζεται το διάγραμμα του μετατροπέα. Το διαμορφωμένο σήμα PAL, προστίθεται με το σήμα από τη γεννήτρια τυχαίων αριθμών. Από το αποτέλεσμα της πρόσθεσης κρατάμε τα 3 σημαντικότερα ψηφία, στα οποία αργότερα γίνεται κωδικοποίηση σε θερμομετρική κλίμακα (thermometer coding). Η έξοδος αυτής της διαδικασίας οδηγείται σε ένα δίκτυο αντιστάσεων και αθροίζεται στην έξοδο



Σχήμα 4.10: Διάγραμμα μετατροπέα ψηφιακού σήματος σε αναλογικό με χρήση λευκού θορύβου

Ο κώδικας για τη μετατροπή του ψηφιακού σήματος σε αναλογικό είναι ο ακόλουθος:

```
//4bit Thermometer Dithering
reg [15:0] Temp;
reg [2:0] MSB;

always @(posedge CLK)
begin
    {MSB,Temp}<=R+AM_SIGNAL;
    if (MSB==4) OUT<=4'b1111;
    else if (MSB==3) OUT<=4'b0111;
    else if (MSB==2) OUT<=4'b0011;
    else if (MSB==1) OUT<=4'b0001;
    else if (MSB==0) OUT<=4'b0000;
end
```

4.2.4.a Γεννήτρια τυχαίων αριθμών (RAND module)

Η γεννήτρια τυχαίων αριθμών είναι απαραίτητη για τη μετατροπή του ψηφιακού σήματος σε αναλογικό. Εσωτερικά η γεννήτρια υλοποιεί δύο καταχωρητές ολίσθησης με γραμμική ανάδραση (LFSR). Οι καταχωρητές αν έχουν αρκετά μεγάλο μήκος, μπορούν να παράξουν μια ψευδοτυχαία ακολουθία δυαδικών αριθμών. Αυτή η ακολουθία δεν είναι πραγματικά τυχαία, αφού επαναλαμβάνεται περιοδικά, αλλά για τις περισσότερες εφαρμογές μπορεί αν θεωρηθεί ιδανικά τυχαία. Ενδεικτικά αναφέρουμε ότι ένας LSFR 63-bit έχει θα κάνει $2^{63}-1$ κύκλους ρολογιού για να επαναληφθεί. Με ένα ρολόι στα 50 MHz, αυτό μεταφράζεται σε μια περίοδο επανάληψης 5,849 χρόνων. Στη προκειμένη περίπτωση χρησιμοποιούμε δύο LFSR, μήκους 21 και 23 bit. Η έξοδος της γεννήτριας τυχαίων αριθμών δίνεται από το λογικό XOR μεταξύ των εξόδων των δύο LSFR. Με αυτή τη τεχνική αυξάνεται κατά πολύ η περίοδος επανάληψης, υλοποιώντας μια πιο αποδοτική γεννήτρια τυχαίων αριθμών.

Κώδικας RAND module

```
`timescale 1ns / 1ps

module RAND #(parameter Size=17)
    (input wire CLK,RST,
     output reg [Size-1:0] DATA);

wire [Size-1:0] LFOUT0,LFOUT1;
LFSR #(21,Size, 'b10100_00000000_00000000)
LF0(LCLK,RST,LFOUT0);
LFSR #(23,Size, 'b1000010_00000000_00000000)
LF1(LCLK,RST,LFOUT1);

integer i;
always @(posedge CLK)
    for (i=0; i<Size; i=i+1)
        DATA[i]<=LFOUT0[i]^LFOUT1[Size-i-1];

Endmodule
```

5

Μετρήσεις, αποτελέσματα και μελλοντικές επεκτάσεις

Σε αυτό το κεφάλαιο συνοψίζονται οι πειραματικές μετρήσεις, αποτελέσματα προσομοιώσεων. Ταυτόχρονα προτείνονται μελλοντικές επεκτάσεις στον παρόντος συστήματος.

5.1 Σύνοψη:

Στόχος της παρούσας διπλωματικής ήταν η ανάπτυξη ενός πλήρως ψηφιακού πομπού αναλογικού σήματος εικόνας στα VHF. Συγκεκριμένα υλοποιήθηκαν αλγόριθμοι για διαχείριση της ψηφιακής κάμερας, υλοποίηση αλγορίθμων κωδικοποίησης αναλογικής τηλεόρασης, διαμόρφωσης πλάτους και μετατροπής ψηφιακού σήματος σε αναλογικό. Η καινοτομία στη σχεδίαση έγκειται στο γεγονός ότι η μετατροπή του σήματος από ψηφιακό σε αναλογικό γίνεται μία νέα τεχνική που ελαχιστοποιεί τη χρήση αναλογικών εξαρτημάτων, καθιστώντας τη σχεδίαση ενός πλήρως ψηφιακού κυκλώματος πιο εύκολη.

Στα πρώτα κεφάλαια δίνονται οι απαραίτητες θεωρητικές βάσεις που χρειάζονται για την κατανόηση των τεχνικών αναλογικής διαμόρφωσης, αλλά και μια εκτενής περιγραφή των πρωτοκόλλων που υλοποιήθηκαν. Ακολούθως παρουσιάζονται οι τεχνικές σχεδίασης και υλοποίησης αλγορίθμων σε υλικό και τέλος παρουσιάζεται η μελέτη και υλοποίηση των αλγορίθμων σε FPGA.

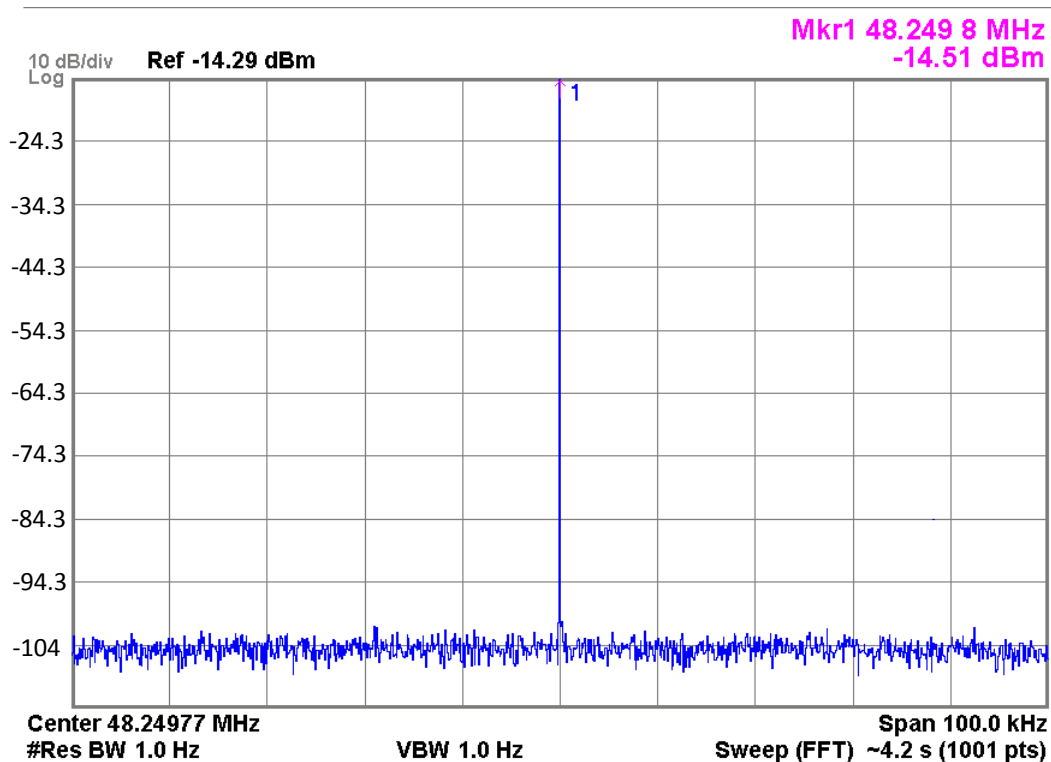
Για τη μελέτη, σχεδίαση και υλοποίηση του αλγορίθμου ακολουθήθηκε η εξής διαδικασία:

- Αρχικά μελετήθηκε το πρωτόκολλο PAL και ορίστηκαν οι περιορισμοί του συστήματος, π.χ. συχνότητα ρολογιού
- Ακολούθησε η σχεδίαση του αλγορίθμου που υλοποιεί τη κωδικοποίηση PAL σε μηχανή πεπερασμένων καταστάσεων και η συγγραφή του κώδικα σε γλώσσα Verilog για τη συγκεκριμένη μηχανή.
- Η σωστή λειτουργία του κώδικα ελέγχθηκε με το λογισμικό modelsim της εταιρίας mentor.
- Ακολούθησε η σύνθεση και υλοποίηση του συστήματος σε FPGA κάνοντας χρήση της πλατφόρμας Xilinx ISE. Το σύστημα προγραμματίστηκε με χρήση του λογισμικού Papilio Loader και έγινε έλεγχος της σωστής λειτουργίας του

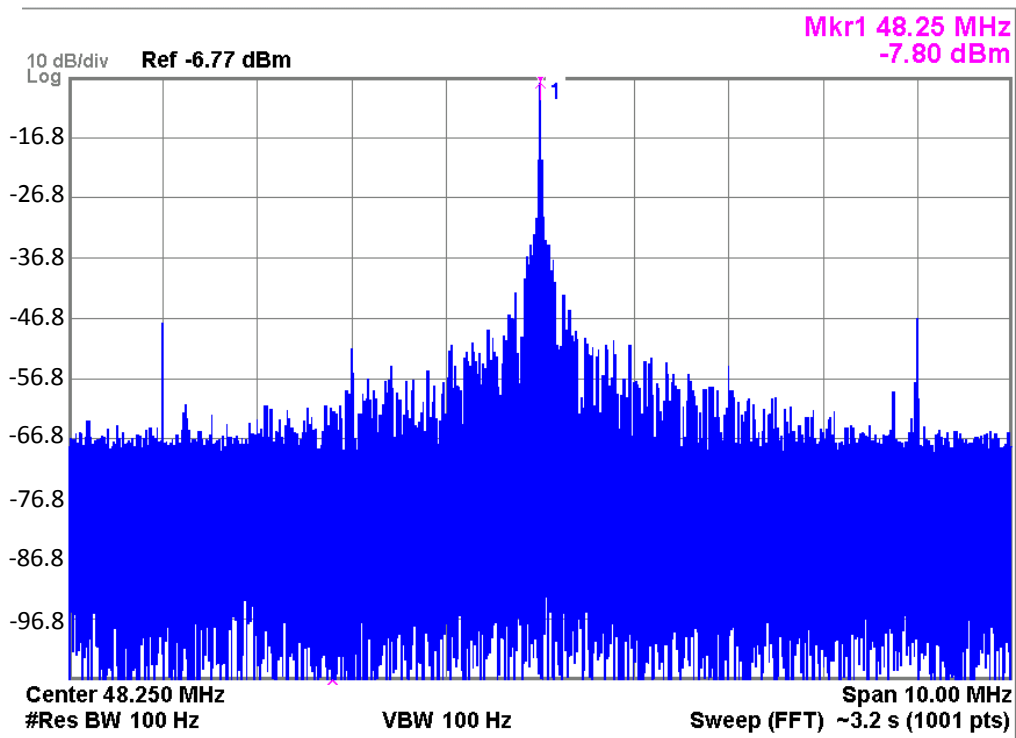
αλγόριθμου, συνδέοντας ενσύρματα τη πλακέτα FPGA με τη θύρα av-in της τηλεόρασης.

- Ακολούθησε η επέκταση του συστήματος, με συγγραφή του αλγορίθμου για χρήση της ψηφιακής κάμερας και υλοποίηση και δοκιμή ακλουθώντας ακριβώς τα ίδια βήματα
- Αφού επιβεβαιώθηκε η λειτουργία του συστήματος ενσύρματα, ακολούθησε η συγγραφή του κώδικα για αναλογική μετάδοση σε επιλεγμένο κανάλι στα VHF. Ακολούθησε προσομοίωση με τα εργαλεία modelsim και iSim των εταιριών mentor και Xilinx.
- Αφού έγινε η σύνθεση και υλοποίηση και προγραμματισμός του FPGA, ακλούθησε η επιβεβαίωση της σωστής λειτουργίας του αλγορίθμου με χρήση φορητού τηλεοπτικού δέκτη, όπως και μετρήσεις σε αναλυτή φάσματος.

Στις παρακάτω εικόνες παρουσιάζονται οι μετρήσεις από τον αναλυτή φάσματος για δύο διαφορετικές περιπτώσεις. Στην πρώτη έχουμε μόνο μετάδοση του ημιτονικού φέροντος στη συχνότητα 48.25 MHz ενώ στη δεύτερη παρουσιάζεται το φάσμα κατά τη μετάδοση του διαμορφωμένου σήματος τηλεόρασης.



Σχήμα 5.1: Φάσμα φέροντος στα 48.25 MHz



Σχήμα 5.2: Φάσμα σήματος Pal διαμορφωμένου στα 48.25 MHz

5.2 Μελλοντικές Επεκτάσεις

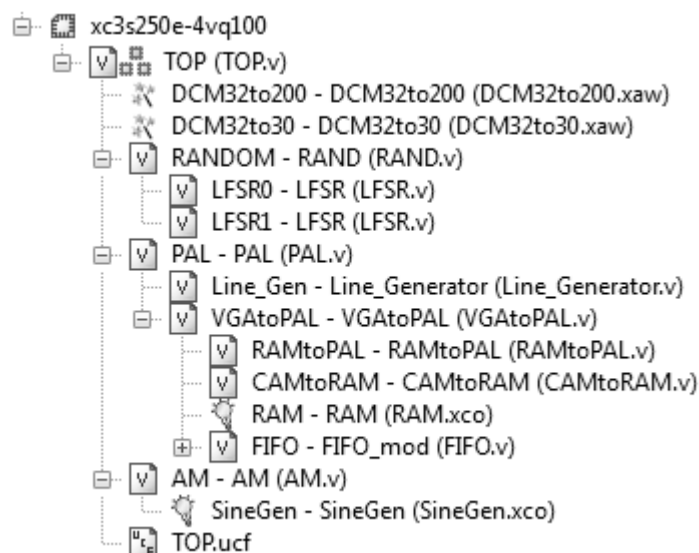
Μερικές από τις μελλοντικές επεκτάσεις αλλά και διορθώσεις του παρόντος αλγορίθμου παρουσιάζονται παρακάτω:

- Επέκταση του συστήματος για κωδικοποίηση έγχρωμου σήματος τηλεόρασης, υλοποιώντας ορθογωνική διαμόρφωση πλάτους (QAM).
- Εξέταση αποδοτικότερων αλγορίθμων για μετατροπή του ψηφιακού σήματος σε αναλογικό.
- Κατασκευή κυκλώματος για ενίσχυση του σήματος εξόδου, με χρήση ενισχυτή κλάσης D.
- Υλοποίηση Ψηφιακού φίλτρου για αποτροπή μετάδοσης του σήματος σε ανεπιθύμητες συχνότητες

Παράρτημα – Κώδικας υλοποίησης σε Verilog

Εδώ παρουσιάζονται όλα τα αρχεία υλοποίησης της υλοποίησης που παρουσιάστηκε στη γλώσσα περιγραφής υλικού Verilog.

Ιεραρχία:



TOP.v

```
`timescale 1ns / 1ps
module TOP(input wire CLKin,RSTin,
           input wire VSYNCin,HREFin,PXL_CLK,
           input wire [7:0] Din,
           output wire OV_RESET,PWDN,XCLK,
           output reg [3:0] OUT);

wire CLK,oCLK,Locked,RST,CLKtoDCM;
wire [8:0] PAL_SIGNAL;
wire PXL_CLK_Buffed;

assign PWDN = 0, OV_RESET = !RST;

BUFG B1(.I(CLKin),.O(CLKtoDCM));
BUFG B2(.I(PXL_CLK),.O(PXL_CLK_Buffed));

//bufs CAM signals
reg VSYNC,HREF;
reg [7:0] D;
always @(posedge PXL_CLK_Buffed)
begin
    VSYNC<=VSYNCin;
    HREF<=HREFin;
    D<=Din;
end
end
```

```

//20Mhz CLK for PAL
wire pCLK;
reg [3:0] cc;

always @(posedge CLK)
    if (RST | cc==4'd9) cc<=4'd0;
    else cc<=cc+4'd1;

BUFG B0(.I(cc[3]),.O(pCLK));

//DCM 32 to 200 MHz
wire CLK_DCM_OUT;
DCM32to200 DCM32to200
(.CLKIN_IN(CLKtoDCM),.RST_IN(RSTin),.CLKFX_OUT(CLK_DCM_OUT),.C
LK0_OUT(oCLK),.LOCKED_OUT(Locked));

//DCM 32 to 30 MHz for camera clock
DCM32to30 DCM32to30
(.CLKIN_IN(CLKtoDCM),.RST_IN(RSTin),.CLKFX_OUT(XCLK),.CLK0_OUT
());
assign RST=~Locked;
BUFG B3(.I(CLK_DCM_OUT),.O(CLK));

wire [15:0] R;
RAND #(16) RANDOM(CLK,RST,R);

//PAL module
PAL PAL(pCLK,RST,PAL_SIGNAL,VSYNC,HREF,PXL_CLK_Buffed,D);

//Amplitude Modulation module
wire [17:0] AM_SIGNAL;
AM AM(CLK,RST,PAL_SIGNAL,AM_SIGNAL);

//4bit Thermometer Dithering
reg [15:0] Temp;
reg [2:0] MSB;
always @(posedge CLK)
    begin
        {MSB,Temp}<=R+AM_SIGNAL;
        if (MSB==4) OUT<=4'b1111;
        else if (MSB==3) OUT<=4'b0111;
        else if (MSB==2) OUT<=4'b0011;
        else if (MSB==1) OUT<=4'b0001;
        else if (MSB==0) OUT<=4'b0000;
    end
endmodule

```

```
-----  
-----  
                                RAND.v  
-----  
-----
```

```
`timescale 1ns / 1ps  
  
module RAND #(parameter Size=17)  
    (input wire CLK,RST,  
     output reg [Size-1:0] DATA);  
  
    wire [Size-1:0] LFOUT0,LFOUT1;  
    LFSR #(21,Size, 'b10100_00000000_00000000)  
    LFSR0 (CLK,RST,LFOUT0);  
    LFSR #(23,Size, 'b1000010_00000000_00000000)  
    LFSR1 (CLK,RST,LFOUT1);  
  
    integer i;  
    always @(posedge CLK)  
        for (i=0; i<Size; i=i+1)  
            DATA[i]<=LFOUT0[i]^LFOUT1[Size-i-1];  
  
endmodule
```

```
-----  
-----  
                                LFSR.v (LFSR0)  
-----  
-----
```

```
`timescale 1ns / 1ps  
  
module LFSR #(parameter Size=8,OSize=6,Mask=8'b1111_0000)  
    (input wire CLK,RST,  
     output [OSize-1:0] OUT);  
  
    reg [Size-1:0] DATA;  
  
    always @(posedge CLK)  
        if (RST) DATA<=~0;  
        else DATA<={DATA[Size-2:0],^(DATA & Mask)};  
  
    assign OUT=DATA[OSize-1:0];  
  
endmodule
```

```

-----
-----
                                LFSR.v (LFSR1)
-----
-----
`timescale 1ns / 1ps

module LFSR #(parameter Size=8, OSize=6, Mask=8'b1111_0000)
    (input wire CLK, RST,
     output [OSize-1:0] OUT);

reg [Size-1:0] DATA;

always @(posedge CLK)
    if (RST) DATA<=~0;
    else DATA<={DATA[Size-2:0],^(DATA & Mask)};

assign OUT=DATA[OSize-1:0];

endmodule

-----
-----
                                PAL.v
-----
-----

module PAL (input wire clk,
            input wire rst,
            output reg [8:0] PAL_signal,
            input wire VSYNC, HREF, PXL_CLK,
            input wire [7:0] D);

parameter [2:0] long_long = 3'b000,
               long_short = 3'b001,
               short_long = 3'b010,
               short_short = 3'b011,
               data_field_1 = 3'b100,
               data_field_2 = 3'b101,
               half_data = 3'b110,
               frame = 3'b111;

reg [2:0] state;
wire [10:0] counter;
wire [9:0] line;

//Line Counter module
Line_Generator Line_Gen(clk, rst, counter, line);

//VGA to PAL module
wire [7:0] PAL_OUT;
VGAtPAL VGAtPAL
(clk, rst, counter, state, VSYNC, HREF, PXL_CLK, D, PAL_OUT);

```

```

reg [8:0] PAL_SIG;
always @(posedge clk) PAL_SIG <= PAL_OUT + 8'd153;

// Next State Logic
always @(posedge clk)
if (rst) state <= long_long;
else
begin
case (state)
long_long: if (line == 2) state <= long_short;
           else if (line == 315) state <= short_short;

long_short : if (line == 3) state <= short_short;

short_short : if (line == 5 || line == 317) state <=
frame;
           else if (line == 0) state <= long_long;
           else if (line == 312) state <= short_long;

short_long  : if (line == 313) state <= long_long;

half_data   : if (line == 622) state <= short_short;

data_field_1: if (line == 277) state <= frame;

data_field_2: if (line == 589) state <= frame;

frame: if (line == 37) state <= data_field_1;
      else if (line == 310) state <= short_short;
      else if (line == 349) state <= data_field_2;
      else if (line == 621) state <= half_data;
endcase
end

//Output Logic
always @ (posedge clk)
case(state)
long_long : if (counter == 0) PAL_signal <= 1;
           else if (counter == 545) PAL_signal <= 153;
           else if (counter == 639) PAL_signal <= 1;
           else if (counter == 1185) PAL_signal <= 153;

long_short: if (counter == 0) PAL_signal <= 1;
           else if (counter == 545) PAL_signal <= 153;
           else if (counter == 639) PAL_signal <= 1;
           else if (counter == 686) PAL_signal <= 153;

short_long: if (counter == 0) PAL_signal <= 1;
           else if (counter == 46) PAL_signal <= 153;
           else if (counter == 639) PAL_signal <= 1;
           else if (counter == 1185) PAL_signal <=
153;

short_short: if (counter == 0) PAL_signal <= 1;
            else if (counter == 46) PAL_signal <= 153;

```

```

else if (counter == 639) PAL_signal <= 1;
else if (counter == 686) PAL_signal <= 153;

half_data : if (counter == 0) PAL_signal <= 153;
else if (counter == 29) PAL_signal <= 1;
else if (counter == 123) PAL_signal <= 153;

//Chrominance Burst Start
else if (counter == 141) PAL_signal <= 153;
//Chrominance Burst End

else if (counter == 186) PAL_signal <= 153;

//Half Data
else if (counter == 239) PAL_signal <= 153;
else if (counter == 369) PAL_signal <= 204;
else if (counter == 449) PAL_signal <= 255;
else if (counter == 629) PAL_signal <= 306;

//Sync Pulse
else if (counter == 639) PAL_signal <= 1;
else if (counter == 686) PAL_signal <= 153;

data_field_1: if (counter == 0) PAL_signal <= 153;
else if (counter == 29) PAL_signal <= 1;
else if (counter == 123) PAL_signal <= 153;

//Chrominance Burst Start
else if (counter == 141) PAL_signal <= 153;
//Chrominance Burst End

else if (counter == 186) PAL_signal <= 153;

//PAL Visible Signal
else if (counter == 239) PAL_signal <= 153;
else if (counter >= 359 & counter <1159)
PAL_signal <= PAL_SIG;
else if (counter == 1159) PAL_signal <= 153;

data_field_2:if (counter == 0) PAL_signal <= 153;
else if (counter == 29) PAL_signal <= 1;
else if (counter == 123) PAL_signal <= 153;

//Chrominance Burst Start
else if (counter == 141) PAL_signal <= 153;
//Chrominance Burst End

else if (counter == 186) PAL_signal <= 153;

//PAL Visible Signal
else if (counter == 239) PAL_signal <= 153;
else if (counter >= 359 & counter <1159)
PAL_signal <= PAL_SIG;
else if (counter == 1159) PAL_signal <= 153;
//800 pixel

```



```

        frame: if (counter == 0) PAL_signal <= 153;
               else if (counter == 29) PAL_signal <= 1;
               else if (counter == 123) PAL_signal <= 153;

               //Chrominance Burst Start
               else if (counter == 141) PAL_signal <= 153;
               //Chrominance Burst End

               else if (counter == 186) PAL_signal <= 153;

               //Black Frame
               else if (counter == 239) PAL_signal <= 153;
    endcase
endmodule

```

```

-----
-----
                                Line_Generator.v
-----
-----

```

```

module Line_Generator(input wire clk,rst,
                     output reg [10:0] counter,
                     output reg [9:0] line);

always@(posedge clk)
  if (rst)
    begin
      counter<=11'd0;
      line<=10'd0;
    end
  else if (counter == 1279)
    begin
      counter <= 0;
      if (line == 624 ) line<=0; else line<=line+10'd1;
    end
  else counter<=counter+11'd1;

endmodule

```

```

-----
-----
                                VGAtPAL.v
-----
-----

```

```

`timescale 1ns / 1ps

module VGAtPAL  (input wire clk,rst,
                 input wire [10:0] counter,
                 input wire [2:0] state,
                 input wire VSYNC,HREF,PXL_CLK,
                 input wire [7:0] D,
                 output wire [7:0] FIFO_OUT);

```

```

wire [14:0] address_a;
wire [14:0] address_b;
wire enable;
wire [7:0] doutb;

RAMtoPAL RAMtoPAL(clk,rst,counter,state,address_b);
CAMtoRAM CAMtoRAM(PXL_CLK,HREF,VSYNC,address_a,enable);
RAM RAM
(.clka(PXL_CLK),.wea(enable),.addra(address_a),.dina(D),.clkb(
clk),.addrb(address_b),.doutb(doutb));

```

```

-----
-----
                                RAMtoPAL.v
-----
-----

```

```

module RAMtoPAL(input wire clk,rst,
                input wire [10:0] counter,
                input wire [2:0] state,
                output reg [14:0] address_b);

reg [9:0] cam_pixel=0;
reg [8:0] cam_line = 0;
reg pxl_en;
reg [2:0] addr_pixel = 0;

always@(posedge clk)
begin
    //Address change enabling starting from position 359
    //for Field 1 and Field 2 states
    if ((state ==4 || state == 5)&& counter == 359) pxl_en <= 1;
    else if ((state ==4 || state == 5)&& counter == 1159) pxl_en
<= 0; //800 pixel horizontal
end

always@(posedge clk )
begin
    //Line Counter
    if (cam_pixel==799 || rst)
    if (cam_line == 479) cam_line <= 0;
    else cam_line <= cam_line +1;

    //Pixel Counter
    if (pxl_en)
    begin
        if (cam_pixel==799) begin cam_pixel <=0;end
        else cam_pixel <= cam_pixel + 1;

        //For address changing every 5 pixels
        if (addr_pixel == 4) addr_pixel <= 0;
        else addr_pixel <= addr_pixel +1 ;
    end
end
end

```

```

always@(posedge clk)
begin
  //Same pixel repeating 5 times
  if (rst | state ==7) address_b <= 15'd0;
  if (pxl_en && addr_pixel == 4)

  //Same line repaeating twice
  if (cam_pixel==799 && cam_line[0] == 0) address_b <=
address_b - 159;
  else if (address_b == 19199) address_b <= 15'd0;
  else address_b <= address_b + 1;

end
endmodule

-----
-----
                          CAMtoRAM.v
-----
-----

module CAMtoRAM(input wire PXL_CLK,HREF,VSYNC,
                 output reg [14:0] address_a,
                 output reg enable);

reg [10:0] pixel=0;
reg [8:0] VGA_line=0;

always @(posedge PXL_CLK)
if (VSYNC)
begin
  pixel <= 10'd0;
  VGA_line <= 9'd0;
end
else if (HREF)
if (pixel == 1279)
begin
  pixel <= 0;
  if (VGA_line==479) VGA_line <= 0; else VGA_line <= VGA_line
+1;
end
else pixel <= pixel +1;

always @(posedge PXL_CLK)
if (VSYNC) address_a <= 0;
else if (HREF & VGA_line[1:0] == 0)
begin
  enable<=(pixel[2:0]==3'd4);
  if (enable)
  if (address_a == 19199) address_a <= 15'd0; else address_a
<= address_a +15'd1;
end

endmodule

```


AM.v


```
`timescale 1ns / 1ps

module AM (input wire clk,rst,
           input wire [8:0] PAL_SIGNAL,
           output reg [17:0] OUT);

wire signed [8:0] sine;

SineGen SineGen (.clk(clk),.sclr(rst),.sine(sine));

reg signed [8:0] sine_b0;
reg [17:0] pr_0;
reg signed [8:0] PAL_Signed_b0;
reg signed [17:0] OUT;

always @(posedge clk)
begin
    sine_b0 <= sine;
    PAL_Signed_b0 <= -{1'b0, PAL_SIGNAL};
end

always @(posedge clk) pr_0 <= sine_b0*PAL_Signed_b0;

always @(posedge clk)
begin
    OUT <= pr_0 + 18'd131072;
end

endmodule
```


TOP.ucf


```
CONFIG PART = XC3S250E-VQ100-4;

NET CLKin PERIOD = 31.25ns HIGH 50% INPUT_JITTER 200 ps;
# 32.00 MHz

NET PXL_CLK PERIOD = 33 ns HIGH 50% INPUT_JITTER 200 ps;
# 30.00 MHz

NET CLKin LOC = "P89" |IOSTANDARD = LVCMOS25;
# 32.00 MHz oscillator

NET HREFin LOC = "P86" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW
= FAST;

NET XCLK LOC = "P84" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW
= FAST;

NET Din<6> LOC = "P79" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW
= FAST;

NET Din<4> LOC = "P70" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW
= FAST;

NET Din<2> LOC = "P67" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW
= FAST;

NET Din<0> LOC = "P65" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW
= FAST;

NET PWDN LOC = "P62" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW =
FAST;

NET VSYNCin LOC = "P85" |IOSTANDARD = LVCMOS25 |DRIVE = 8
|SLEW = FAST;

NET PXL_CLK LOC = "P83" |IOSTANDARD = LVCMOS25;

NET Din<7> LOC = "P78" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW
= FAST;

NET Din<5> LOC = "P71" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW
= FAST;
```

```
NET Din<3> LOC = "P68" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW  
= FAST;
```

```
NET Din<1> LOC = "P66" |IOSTANDARD = LVCMOS25 |DRIVE = 8 |SLEW  
= FAST;
```

```
NET OV_RESET LOC = "P63" |IOSTANDARD = LVCMOS25 |DRIVE = 8  
|SLEW = SLOW;
```

```
NET RSTin LOC = "P25" |IOSTANDARD = LVTTTL |DRIVE = 8 |SLEW =  
SLOW;
```

```
NET OUT<3> LOC = "P5" |IOSTANDARD = LVCMOS33 |DRIVE = 16 |SLEW  
= FAST;
```

```
NET OUT<2> LOC = "P9" |IOSTANDARD = LVCMOS33 |DRIVE = 16 |SLEW  
= FAST;
```

```
NET OUT<1> LOC = "P10" |IOSTANDARD = LVCMOS33 |DRIVE = 16  
|SLEW = FAST;
```

```
NET OUT<0> LOC = "P11" |IOSTANDARD = LVCMOS33 |DRIVE = 16 |SLEW  
= FAST;
```

Βιβλιογραφία

- [1] K. Vasiliou, K. Galanopoulos, P. Sotiriadis, "All-Digital Video RF Transmitter with Embedded Direct Frequency Synthesizer And an FPGA implementation of It", IEEE International Frequency Control Symposium 2013.
- [2] Γεώργιος Κ. Καραγιαννίδης, Τηλεπικοινωνιακά συστήματα 2η Έκδοση, Αθήνα: Εκδόσεις Τζιόλα 2010
- [3] ITU Radio Regulations, Volume 1, Article 2; Edition of 2008
- [4] Simon Haykin, Συστήματα Επικοινωνίας, Αθήνα: Εκδόσεις Παπασωτηρίου, 1995
- [5] Παναγιώτης Γ. Κωστής, Εισαγωγή στις τηλεπικοινωνίες : διαμόρφωση και μετάδοση σημάτων, Αθήνα : Τζιόλας , 2012
- [6] History of Television, Ανάκτηση June 2013 από:
<http://www2.hesston.edu/Physics/TelevisionDisplays/HISTORY1.HTM>
- [7] NTSC, Ανάκτηση June 2013 από Wikipedia:
<http://en.wikipedia.org/wiki/NTSC>
- [8] World Analogue Television Standards and Waveforms, Ανάκτηση June 2013 από Pembers' Ponderings:
<http://www.pembers.freemove.co.uk/World-TV-Standards/Colour-Standards.html#SECAM-IV>
- [9] Rec. ITU-R BT.470-5 1 Recommendation ITU-R BT.470-5* CONVENTIONAL TELEVISION SYSTEMS Available at: http://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.470-5-199802-S!!PDF-E.pdf
- [10] John Watkinson, The Engineer's Guide to Decoding & Encoding, Hampshire, UK: Snell & Wilcox
- [11] Rafael C. Gonzalez, Richard E. Woods, Ψηφιακή Επεξεργασία Εικόνας 3^η Έκδοση, Αθήνα: Εκδόσεις Τζιόλα, 2011
- [12] OV7670/OV7171 CMOS VGA (640x480) CameraChip™ Implementation Guide από:
<http://www.robokits.co.in/datasheets/OV7670.pdf>
- [13] Field-programmable gate array. Ανάκτηση June 2013, από Wikipedia:
https://en.wikipedia.org/wiki/Field-programmable_gate_array
- [14] Κιαμάλ Πεκμεστζή, Παρουσίαση: Βασικά στοιχεία σχεδίασης κυκλωμάτων CMOS – VLSI, Ανάκτηση June 2013 από microlab.ntua.gr:
<http://courses.microlab.ntua.gr/mod/resource/view.php?id=1012>
- [15] Spartan-3E FPGA Family Data Sheet, Ανάκτηση June 2013, από Xilinx:
http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf
- [16] K. Galanopoulos, P. Sotiriadis, "Modulation Techniques for All-Digital Transmitters Based on Pulse Direct Digital Synthesizers", IEEE International Frequency Control Symposium 2012.
- [17] LogiCORE IP Block Memory Generator v6.1, Ανάκτηση June 2013 από Xilinx:
http://www.xilinx.com/support/documentation/ip_documentation/blk_mem_gen/v7_3/pg058-blk-mem-gen.pdf
- [18] P. Sotiriadis, K. Galanopoulos*, "Direct All-Digital Frequency Synthesis Techniques, Spurs Suppression, and Deterministic Jitter Correction", IEEE Trans. on Circuits and Systems-I, Vol. 59, No. 5, May 2012, pp. 958-968.
- [19] K. Galanopoulos, P. Sotiriadis, "Optimal Dithering Sequences for Spurs Suppression in Pulse Direct Digital Synthesizers", IEEE International Frequency Control Symposium, 2012.
- [20] P. Sotiriadis, N. Miliou, "Single-Bit Digital Frequency Synthesis via Dithered Nyquist-rate Sinewave Quantization", to appear in the IEEE Trans. on Circuits and Systems-I