



---

Σχεδιασμός και υλοποίηση μεθόδων  
Intra πρόβλεψης για το H.264

Διπλωματική εργασία:

*Π. Ζηλαράκης*

Πανεπιστήμιο Θεσσαλίας  
Τμήμα Μηχανικών Η/Υ , Τηλ & Δικτύων

Οκτώβριος , 2007

---



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΒΙΒΛΙΟΘΗΚΗ & ΚΕΝΤΡΟ ΠΛΗΡΟΦΟΡΗΣΗΣ  
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»**

Αριθ. Εισ.: 5987/1  
Ημερ. Εισ.: 29-10-2007  
Δωρεά: Συγγραφέα  
Ταξιθετικός Κωδικός: ΠΤ – ΜΗΥΤΔ  
2007  
ΣΜΑ

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή του τμήματος Μηχανικών Η/Υ Τηλεπικοινωνιών και Δικτύων και βασικό επιβλέποντα της πτυχιακής αυτής εργασίας κ.Γεώργιο Σταμούλη που μου έδωσε την ευκαιρία να πραγματοποιήσω αυτή την μελέτη. Η υποστήριξη του συντέλεσε στην έγκαιρη και έγκυρη ολοκλήρωση αυτής της μελέτης.

Επιπρόσθετα, θα ήθελα να ευχαριστήσω την διδακτορική φοιτήτρια του τμήματος Κοζύρη Μαρία για τις εύστοχες υποδείξεις της σε θεωρητικά ζητήματα και για την καθοδήγηση της κατά τη διάρκεια της διπλωματικής εργασίας. Ακόμη θα ήθελα να ευχαριστήσω το διδακτορικό φοιτητή του τμήματος Δαδαλιάρη Αντώνιο για την ανεκτίμητη βοήθεια του στην εκμάθηση χρήσιμων εργαλείων που συντέλεσαν στην εκπόνηση της διπλωματικής εργασίας και για τις παρατηρήσεις του σε ζητήματα υλοποίησης.

Θα ήθελα να ευχαριστήσω επίσης όλους τους μεταπτυχιακούς και διδακτορικούς φοιτητές του γραφείου Ε5, φίλους και γνωστούς για την βοήθεια που μου παρείχαν όλο αυτό το διάστημα.

Τέλος, ένα πολύ μεγάλο ευχαριστώ στην οικογένεια μου που με στήριξε με όλους τους δυνατούς τρόπους αυτά τα πέντε χρόνια στο Πανεπιστήμιο Θεσσαλίας, στο τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών & Δικτύων ως προπτυχιακό φοιτητή.

# Παράρτημα 1

1. Τι είναι το H.264
2. Πώς ένας κωδικοποιητής –αποκωδικοποιητής H.264 λειτουργεί
  - 2.1 Διαδικασία Κωδικοποιητή
    - 2.1.1 Πρόβλεψη
    - 2.2.2 Μετατροπή και κβαντοποίηση
    - 2.1.3 Κωδικοποίηση bitstream
  - 2.2 Διαδικασία Αποκωδικοποιητή
    - 2.2.1 Αποκωδικοποίηση bitstream
- 2.2.2 Κλιμάκωση και αντίστροφη μετατροπή
- 2.2.3 Αναδημιουργία

1. Δομή H.264
  - 1.1 Profiles
  - 1.2 Μορφή βίντεο
  - 1.3 Μορφή κωδικοποιημένων δεδομένων
  - 1.4 Φέτες (Slices )
    - 1.4.1 I καρέ (intra frames)
    - 1.4.2 P καρέ (Predicted frames)
    - 1.4.3 B καρέ (Bi- directional frames)
    - 1.4.4 SP & SI καρέ (SP και SI frames)
2. Η διαδικασία Κωδικοποίησης –Αποκωδικοποίησης
  - 2.1 Διαδικασία κωδικοποίησης
  - 2.2 Κωδικοποιητής
    - 2.2.1 Κωδικοποιητής (κατεύθυνση κωδικοποίησης)
    - 2.2.2 Κωδικοποιητής (κατεύθυνση αναδημιουργίας)
  - 2.3 Αποκωδικοποιητής
3. Το H.264 στην πράξη
  - 3.1 Απόδοση
  - 3.2 Εφαρμογές
  - 3.3 Κοινά στοιχεία με άλλα πρότυπα

- 1 . Πρόβλεψη των intra macroblocks
- 2 . Τρόποι πρόβλεψης 4X4 μπλοκ φωτεινότητας
  - 2.1 Κάθετη Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Vertical )
  - 2.2 Οριζόντια Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Horizontal )
  - 2.3 Πρόβλεψη Μέσου όρου μπλοκ φωτεινότητας 4X4 (DC )
  - 2.4 Διαγώνια Κάτω αριστερά Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Diagonal Down –Left )
  - 2.5 Διαγώνια Κάτω δεξιά Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Diagonal Down- Right )
  - 2.6 Κάθετη δεξιά Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Vertical Right )
  - 2.7 Οριζόντια Κάτω Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Horizontal-Down )
  - 2.8 Κάθετη Αριστερά Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Vertical-Left)
  - 2.9 Πάνω οριζόντια Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Horizontal- Up)
- 2.10 Χαρακτηριστικό παράδειγμα
- 3 . Τρόποι πρόβλεψης 16X16 μπλοκ φωτεινότητας
  - 3.1 Κάθετη Πρόβλεψη μπλοκ φωτεινότητας 16X16 (Vertical )
  - 3.2 Οριζόντια Πρόβλεψη μπλοκ φωτεινότητας 16X16 (Horizontal)
  - 3.3 Πρόβλεψη μέσου όρου μπλοκ φωτεινότητας 16X16 (DC )
  - 3.4 Plane Πρόβλεψη μπλοκ φωτεινότητας 16X16 (Plane )
  - 3.5 Χαρακτηριστικό παράδειγμα
- 4 . Intra prediction 4X4 vs. Intra Prediction 16X16
- 5 . Τρόποι πρόβλεψης 8X8 μπλοκ χρώματος
- 6 . Επιλογή κατάλληλου τρόπου

- 1 . Εισαγωγή
- 2 . Σχεδίαση τρόπων πρόβλεψης 4X4 μπλοκ φωτεινότητας
  - 2.1 Μαθηματική αναπαράσταση τρόπων πρόβλεψης 4X4 μπλοκ φωτεινότητας σύμφωνα με το πρότυπο H.264 και Τροποποίηση
    - 2.1.1 Κάθετη Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Vertical )
    - 2.1.2 Οριζόντια Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Horizontal )
    - 2.1.3 Πρόβλεψη Μέσου όρου μπλοκ φωτεινότητας 4X4 (DC )

2.1.4 Διαγώνια Κάτω αριστερά Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Diagonal Down –Left )	
2.1.5 Διαγώνια Κάτω δεξιά Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Diagonal Down- Right )	
2.1.6 Κάθετη δεξιά Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Vertical-Right )	
2.1.7 Οριζόντια Κάτω Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Horizontal-Down )	
2.1.8 Κάθετη Αριστερά Πρόβλεψη μπλοκ φωτεινότητας 4X4 (Vertical-Left)	
2.1.9 Πάνω οριζόντια Πρόβλεψη μπλοκ φωτεινότητας 4X4	
3 . Σχεδίαση τρόπων πρόβλεψης 16X16 μπλοκ φωτεινότητας	
3.1 Μαθηματική αναπαράσταση τρόπων πρόβλεψης 16X16 μπλοκ φωτεινότητας σύμφωνα με το πρότυπο H.264 και τροποποίηση	
3.1.1 Κάθετη Πρόβλεψη μπλοκ φωτεινότητας 16X16	
3.1.2 Οριζόντια Πρόβλεψη μπλοκ φωτεινότητας 16X16	
3.1.3 Πρόβλεψη μέσου όρου μπλοκ φωτεινότητας 16X16	
3.1.4 Plane Πρόβλεψη μπλοκ φωτεινότητας 16X16	
4. Σχεδίαση τρόπων πρόβλεψης 4X4 μπλοκ φωτεινότητας	63
1 . Εισαγωγή	64
2 . Υλοποίηση μεθόδων πρόβλεψης για τα μπλοκ φωτεινότητας 4X4	68
3 . Υλοποίηση μεθόδων πρόβλεψης για τα μπλοκ φωτεινότητας 16X16	74
5. Σχεδίαση τρόπων πρόβλεψης 16X16 μπλοκ φωτεινότητας	85
1 . Εισαγωγή	86
2. Design compiler	87
3. Πειραματικές μετρήσεις και αποτελέσματα	91
6. Συμπεράσματα	102



*Η εργασία αυτή είναι  
αφιερωμένη στην οικογένεια  
μου.*



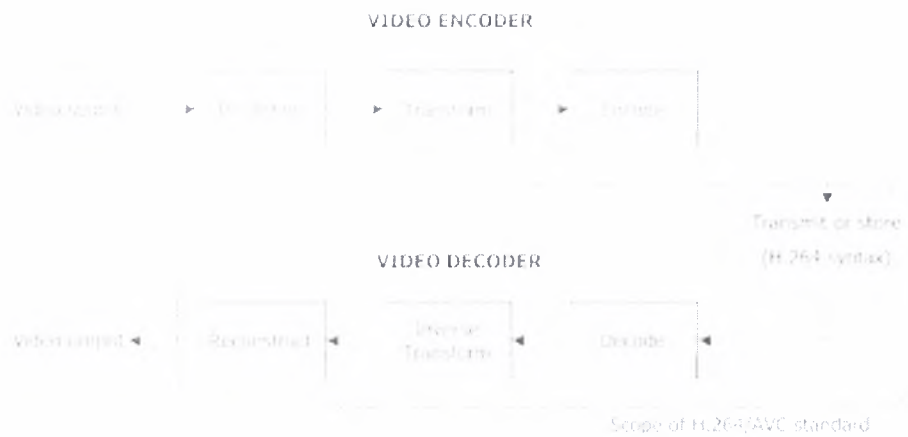
## 1. Τι είναι το H.264

Το h.264 είναι ένα βιομηχανικό πρότυπο (standard) για την συμπίεση βίντεο, δηλαδή την διαδικασία κατά την οποία μετατρέπεται ψηφιακό βίντεο σε μια άλλη μορφή η οποία καταλαμβάνει λιγότερη χωρητικότητα όταν αποθηκεύεται ή μεταδίδεται. Η συμπίεση βίντεο (video coding) είναι μια απαραίτητη τεχνολογία για εφαρμογές όπως

- ❖ ψηφιακή τηλεόραση
- ❖ DVD-video
- ❖ mobile TV
- ❖ video conferencing
- ❖ internet video streaming

Το νέο αυτό πρότυπο έχει τίτλο Advanced Video Coding (AVC) και εκδίδεται από την International standard bodies ITU-T (International Telecommunication Union) και ISO/IEC (International Organization for standardization /International Electron technical Commission). Ορίζει μια σύνταξη για κωδικοποίηση βίντεο και μια μέθοδο για αποκωδικοποίηση αυτής της σύνταξης για να παράγει μια ακολουθία βίντεο που θα μπορεί να προβληθεί. Το πρότυπο αυτό δεν προσδιορίζει επακριβώς πώς να κωδικοποιείται (συμπιέζεται) βίντεο -αυτό αφήνεται στο κατασκευαστή του κωδικοποιητή βίντεο -αλλά στην πράξη τα βήματα που λαμβάνουν χώρα στον κωδικοποιητή είναι τα αντίθετα από αυτά συμβαίνουν κατά την διαδικασία αποκωδικοποίησης. Ένας κωδικοποιητής αναλαμβάνει να μετατρέψει βίντεο σε μια συμπιεσμένη μορφή και ένας αποκωδικοποιητής μετατρέπει συμπιεσμένο βίντεο πίσω στη μη συμπιεσμένη μορφή. Το *σχήμα 1.1* απεικονίζει την διαδικασία κωδικοποίησης και αποκωδικοποίησης.

Το H.264/AVC πρώτα δημοσιεύτηκε το 2003. Χτίστηκε πάνω σε παλαιότερα πρότυπα όπως το MPEG-2 και MPEG-4 και προσφέρει την δυνατότητα για καλύτερη αιόδοση συμπίεσης και μεγαλύτερη ευελιξία στην μεταφορά, στη συμπίεση και στην αποθήκευση βίντεο.

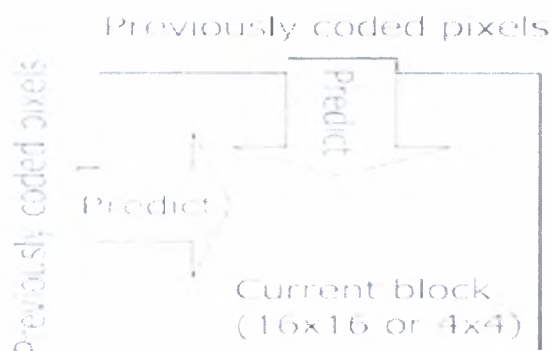


## 2. Πώς ένας κωδικοποιητής-αποκωδικοποιητής H.264 λειτουργεί;

Ένας κωδικοποιητής βίντεο H.264 πραγματοποιεί τις διαδικασίες πρόβλεψης, μετατροπής και κωδικοποίησης (αρχίμορφ) για να παραγάγει ένα συμπίεσμένο H.264 bitstream. Ένας αποκωδικοποιητής βίντεο H.264 πραγματοποιεί τις συμπληρωματικές διαδικασίες της αποκωδικοποίησης, της αντίστροφης μετατροπής και της αναδημιουργίας για να παράγει μια αποκωδικοποιημένη ακολουθία βίντεο.

Ο κωδικοποιητής διαμερίζει ένα καρέ βίντεο στις μονάδες ενός macroblock 16x16 δειγμάτων. Διαμορφώνει μια πρόβλεψη του macroblock αυτού βασισμένη σε στοιχεία που έχουν κωδικοποιηθεί προηγουμένως, είτε από το τρέχον καρέ (intra prediction) είτε από άλλα καρέ που έχουν ήδη κωδικοποιηθεί και μεταδοθεί (inter prediction). Ο κωδικοποιητής αφαιρεί την πρόβλεψη από το τρέχον μπλοκ (μπλοκ 16x16 ή 4x4) για να διαμορφώσει το μπλοκ υπολοίπων (residual block). Οι μέθοδοι πρόβλεψης που υποστηρίζονται από το H.264 είναι πιο εύκαμπτες από εκείνες στα προηγούμενα πρότυπα, επιτρέποντας περισσότερο ακριβείς προβλέψεις και ως εκ τούτου αποδοτικότερη συμπίεση βίντεο.

Η πρόβλεψη χρησιμοποιεί 16x16 και 4x4 μεγέθη μπλοκ για να προβλέψει το macroblock από γειτονικά δείγματα που έχουν κωδικοποιηθεί προηγουμένως μέσα στο ίδιο καρέ (αρχικά 0.2). Η πρόβλεψη χρησιμοποιεί μια σειρά μεγεθών μπλοκ (από 16x16 έως 4x4) για να προβλέψει τα δείγματα στο τρέχον καρέ από παρόμοιες περιοχές προηγουμένων κωδικοποιημένων καρέ (αρχικά 0.3).

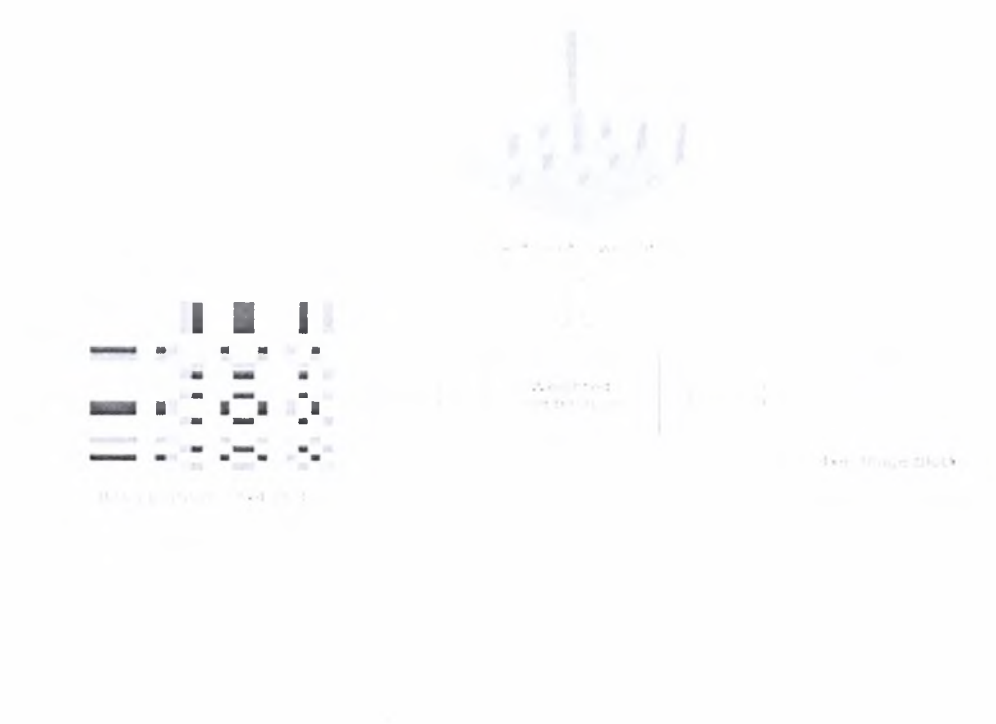




Ένα μπλοκ με τα υπολειπόμενα δείγματα (residual block) μετασχηματίζεται χρησιμοποιώντας 4x4 ή 8x8 integer transform, μια προσεγγιστική μορφή του διακριτού μετασχηματισμού συνημίτονου (DCT). Η μετατροπή παράγει ένα σύνολο συντελεστών, κάθε ένας από τους οποίους είναι μια βεβαρημένη τιμή για κάθε σταθμισμένο σχέδιο βάσης. Όταν συνδυάζονται, τα σταθμισμένα σχέδια βάσης αναδημιουργούν το μπλοκ με τα υπολειπόμενα δείγματα. Το [εξήγησε πώς](#) επιδεικνύει πώς ο αντίστροφος DCT δημιουργεί έναν μπλοκ εικόνας με τη στάθμιση κάθε σχεδίου βάσης σύμφωνα με μια τιμή ενός συντελεστή και το συνδυασμό των βεβαρημένων σχεδίων βάσης.

Το αποτέλεσμα της μετατροπής ενός μπλοκ με μετασχηματισμένους συντελεστές, κβαντοποιείται, δηλαδή κάθε συντελεστής διαιρείται με μια ακέραια τιμή. Η κβαντοποίηση μειώνει την ακρίβεια των συντελεστών μετατροπής σύμφωνα με μια παράμετρο κβαντοποίησης (QP). Τυπικά το αποτέλεσμα είναι ένα μπλοκ στο οποίο οι περισσότεροι ή όλοι οι συντελεστές είναι μηδέν, με μερικούς συντελεστές μη μηδενικούς. Ρύθμιση του QP σε μεγάλη τιμή σημαίνει ότι οι περισσότεροι συντελεστές παίρνουν τιμή μηδέν με συνέπεια την υψηλή συμπίεση εις βάρος της φτωχής αποκωδικοποιημένης ποιότητας εικόνας. Θέτοντας QP σε μια χαμηλή τιμή σημαίνει ότι περισσότεροι διαφορετικοί από το μηδέν συντελεστές παραμένουν μετά τον

κβαντισμό με συνέπεια την καλύτερη αποκωδικοποιημένη ποιότητα εικόνας αλλά τη χαμηλότερη συμπίεση.



Η διαδικασία κωδικοποίησης βίντεο παράγει διάφορες τιμές που πρέπει να κωδικοποιηθούν για να διαμορφώσουν το συμπιεσμένο bitstream. Αυτές οι τιμές περιλαμβάνουν

- ❖ κβαντοποιημένους συντελεστές μετατροπής
- ❖ πληροφορίες για να επιτρέψει στον αποκωδικοποιητή να αναδημιουργήσει την πρόβλεψη
- ❖ πληροφορίες για τη δομή των συμπιεσμένων στοιχείων και τα εργαλεία συμπίεσης που χρησιμοποιούνται κατά τη διάρκεια της κωδικοποίησης
- ❖ πληροφορίες για την πλήρη ακολουθία βίντεο.

Αυτές οι τιμές και οι παράμετροι (στοιχεία σύνταξης) μετατρέπονται σε δυαδικούς κώδικες χρησιμοποιώντας κωδικοποίηση μεταβλητού μήκους (CAVLC) ή/και αριθμητική κωδικοποίηση (CABAC). Κάθε μια από αυτές τις μεθόδους κωδικοποίησης παράγουν μια αποδοτική, συμπαγή δυαδική αναπαράσταση της πληροφορίας. Η κωδικοποιημένη ακολουθία μπορεί έπειτα να αποθηκευτεί ή/και να μεταδοθεί.

Ένας αποκωδικοποιητής βίντεο λαμβάνει το συμπιεσμένο H.264 bitstream, αποκωδικοποιεί κάθε ένα από τα συντακτικά στοιχεία και εξάγει τις πληροφορίες που περιγράφονται παραπάνω (κβαντοποιημένοι συντελεστές, πληροφορίες πρόβλεψης, κλπ). Αυτές οι πληροφορίες χρησιμοποιούνται έπειτα για να αντιστρέψουν τη διαδικασία κωδικοποίησης και αναδημιουργούν μια ακολουθία τηλεοπτικών εικόνων.

Οι κβαντοποιημένοι συντελεστές κλιμακώνονται ξανά. Κάθε συντελεστής πολλαπλασιάζεται με μια ακέραια τιμή για να αποκαταστήσει την αρχική κλιμάκωση του. Μια αντίστροφη μετατροπή συνδυάζει τα τυποποιημένα σχέδια βάσης, που σταθμίζονται από τους επανακλιμακούμενους συντελεστές, για να αναδημιουργήσουν κάθε επιμέρους μπλοκ από τα υπολειπόμενα στοιχεία (residual block). Αυτά τα μπλοκ συνδυάζονται μαζί για να διαμορφώσουν το residual macroblock.

Για κάθε μπλοκ, ο αποκωδικοποιητής διαμορφώνει μια ίδια πρόβλεψη με αυτή που δημιουργείται από τον κωδικοποιητή. Ο αποκωδικοποιητής προσθέτει την πρόβλεψη στο αποκωδικοποιημένο μπλοκ υπολοίπων (residual block) για να αναδημιουργήσει τελικά ένα αποκωδικοποιημένο macroblock που μπορεί έπειτα να επιδειχθεί ως τμήμα ενός τηλεοπτικού πλαισίου.

## 1. Λογική H.264

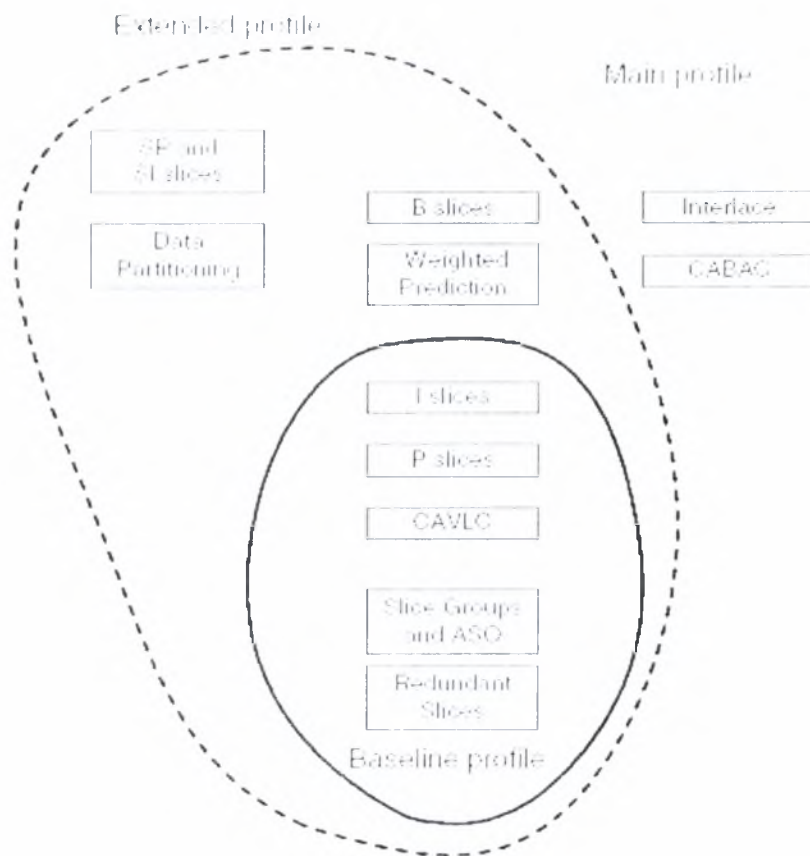
Το H.264 καθορίζει ένα σύνολο τριών Profiles όπου κάθε ένα από αυτά υποστηρίζει ένα ιδιαίτερο σύνολο λειτουργιών κωδικοποίησης και διευκρινίζει τι απαιτείται από έναν κωδικοποιητή ή έναν αποκωδικοποιητή που πειθαρχεί στο συγκεκριμένο profile (σχήμα 1.1).

Το *Baseline Profile* υποστηρίζει intra, inter πρόβλεψη (χρησιμοποιώντας τις I-φέτες και τις P-φέτες) και κωδικοποίηση εντροπίας με τους προσαρμοστικούς κώδικες μεταβλητού-μήκους για ένα καρέ (CAVLC). Οι πιθανές εφαρμογές του συγκεκριμένου profile περιλαμβάνουν τη βιντεοτηλεφωνία, συνεδρίαση μέσω video και ασύρματες επικοινωνίες.

Το *Main Profile* περιλαμβάνει την υποστήριξη για interlaced βίντεο, για inter κωδικοποίηση χρησιμοποιώντας τις B-φέτες, για inter κωδικοποίηση χρησιμοποιώντας τη σταθμισμένη πρόβλεψη και κωδικοποίηση εντροπίας που χρησιμοποιεί την αριθμητική κωδικοποίηση βασισμένη σε καρέ (CABAC). Οι πιθανές εφαρμογές του συγκεκριμένου profile περιλαμβάνουν τηλεοπτική ραδιοφωνική αναμετάδοση και τηλεοπτική αποθήκευση.

Το *Extended Profile* δεν υποστηρίζει το πεπλεγμένο βίντεο ή CABAC αλλά προσθέτει τρόπους για να επιτρέψει την αποδοτική εναλλαγή μεταξύ κωδικοποιημένων bitstream (SP και SI-φέτες) και τη βελτιωμένη ανθεκτικότητα λάθους. Το Extended Profile μπορεί να είναι ιδιαίτερα χρήσιμο για εφαρμογές μέσω ροής.

Εντούτοις, κάθε σχεδιάγραμμα έχει την ικανοποιητική ευελιξία να υποστηρίξει ένα ευρύ φάσμα εφαρμογών.

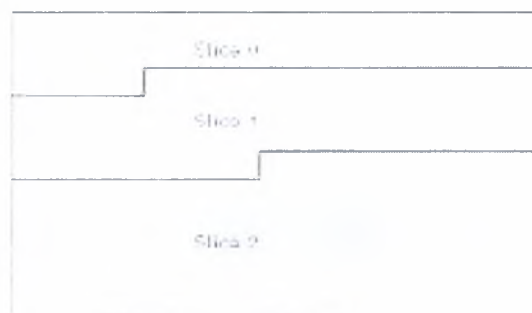


Το H.264 υποστηρίζει την κωδικοποίηση και την αποκωδικοποίηση για 4:2:0 interlaced βίντεο και την προεπιλεγμένη μορφή δειγματοληψίας για 4:2:0 προοδευτικά καρέ. Με το προεπιλεγμένο σχήμα δειγματοληψίας, τα δείγματα χρώματος (Cb και Cr) ευθυγραμμίζονται οριζόντια με κάθε δεύτερο δείγμα φωτεινότητας και τοποθετούνται κάθετα μεταξύ δύο δειγμάτων φωτεινότητας. Ένα πλεγμένο καρέ αποτελείται από δύο τομείς που χωρίζεται εγκαίρως.



Το H.264 κάνει μια διάκριση μεταξύ ενός στρώματος τηλεοπτικής κωδικοποίησης (VCL) και ενός στρώματος αφαίρεσης δικτύου (NAL). Η έξοδος της διαδικασίας κωδικοποίησης είναι δεδομένα VCL (μια ακολουθία από bits τα οποία αναπαριστούν τα κωδικοποιημένα δεδομένα βίντεο) τα οποία χαρτογραφούνται στις μονάδες NAL πριν από τη μετάδοση ή την αποθήκευση.

Ένα καρέ βίντεο κωδικοποιείται σαν μια ή περισσότερες φέτες η καθεμία εκ των οποίων αποτελείται από ένα ή περισσότερα macroblock. Υπάρχει το ενδεχόμενο να υπάρχει ένα macroblock ανά καρέ ή ακόμη και μία φέτα ανά καρέ. Ο αριθμός των macroblocks ανά φέτα μπορεί να μην είναι συνεχόμενος μέσα σε ένα καρέ [\(εικόνα 1.14\)](#)



Υπάρχουν 5 διαφορετικές φέτες και κατ' επέκταση καρέ.

- ❖ **I-καρέ:** Αυτόνομα κωδικοποιημένα καρέ(κωδικοποιημένα ως εικόνες)
- ❖ **P-καρέ:** Κωδικοποιημένα με βάση προβλεπτική κωδικοποίηση από άλλα P ή I καρέ
- ❖ **B-καρέ:** Παρεμβλλόμενα (Interpolation) frames ανάμεσα σε P και I frames

- ❖ Τα καρέ SP και SI επιτρέπουν την αποδοτική μετατροπή μεταξύ των τηλεοπτικών ρευμάτων και της αποδοτικής τυχαίας προσπέλασης για τους τηλεοπτικούς αποκωδικοποιητές.

Το είδος αυτό των καρέ κάνει χρήση του intra frame-coding. Τα καρέ τύπου I είναι τα μόνα που είναι κωδικοποιημένα στο σύνολό τους και η αποκωδικοποίηση μπορεί να γίνει χωρίς αναφορά σε κάποιο άλλο. Αποτελούν σημεία αναφοράς κατά την τυχαία προσπέλαση ενός σήματος. Επειδή η παρουσία τους είναι απαραίτητη σε σημείο χρονικής αναφοράς και για να αποφευχθεί η διάδοση των σφαλμάτων που δημιουργούν τα P καρέ επιβάλλεται να μεταδίδονται ανά τακτά χρονικά καρέ. Έτσι υιάρχει ένα I καρέ τουλάχιστον κάθε 15 καρέ (δηλαδή δύο φορές το δευτερόλεπτο αν η συχνότητα είναι 30 Hz). Η διαδικασία της κωδικοποίησης ενός I καρέ βασίζεται στη λογική του προτύπου JPEG. Η εικόνα χωρίζεται σε macroblocks και για κάθε μπλοκ ξεχωριστά εφαρμόζεται DCT, κβαντοποίηση, Zig-Zag Scanning, Run-Length Encoding και Huffman Encoding.

Τα καρέ τύπου P είναι βασισμένα σε ένα προηγούμενο I ή P καρέ. Με τη βοήθεια του motion compensation προβλέπουν τη νέα θέση όποιων macroblocks έχουν απλά μετακινηθεί και κωδικοποιούν τον αριθμό του macroblock, ένα διάνυσμα κίνησης και τη διαφορά (σφάλμα πρόβλεψης). Με τη σειρά τους μπορούν να αποτελέσουν και αυτά σημείο αναφοράς για επόμενα καρέ και αυτός είναι και ο λόγος που συμβάλλουν στην εισαγωγή και διάδοση σφαλμάτων, αφού η διαδικασία της πρόβλεψης κίνησης δεν μπορεί να είναι 100% ακριβής. Δεν έχουν το μέγεθος των I καρέ γιατί δεν έχουν περιγραφεί με την ίδια ακρίβεια, δηλαδή παρουσιάζουν μεγαλύτερο ποσοστό συμπίεσης. Η διαδικασία λοιπόν της κωδικοποίησης τους περιλαμβάνει:

- ❖ σύγκριση macroblocks του τρέχοντος καρέ με αυτά του προηγούμενου και εύρεση της βέλτιστης ομοιότητας (motion estimation)
- ❖ δημιουργία διανύσματος κίνησης, και υπολογισμός των διαφορών πρόβλεψης (motion compensation)
- ❖ μετασχηματισμός DCT των διαφορών σε κάθε μπλοκ του νέου macroblock , κβαντοποίηση
- ❖ Run-Length-Encoding και το αποτέλεσμα κωδικοποιείται με κωδικοποίηση Huffman, όπως και στα I καρέ

Τα καρέ τύπου B είναι καρέ που δημιουργούνται λαμβάνοντας το μέσο όρο σε επίπεδο macroblock ενός προηγούμενου και ενός επόμενου καρέ I και P (ένα από το κάθε είδος). Δε συντελούν τόσο πολύ στη διάδοση των σφαλμάτων γιατί δεν χρησιμοποιούνται ως σημεία αναφοράς και επιπλέον μειώνουν σημαντικά το σφάλμα παίρνοντας το μέσο όρο από δύο καρέ. Μπορούμε να πούμε ότι ο 'κύκλος της ζωής' τους περιορίζεται μόνο σε αυτά και δεν επεκτείνεται με το να κληροδοτούν πληροφορίες σε άλλα καρέ, κάτι που πολλές φορές σε συνδυασμό και με την υπολογιστική πολυπλοκότητα που απαιτούν για την κωδικοποίηση και αποκωδικοποίηση τα κάνει μη επιθυμητά από τους κατασκευαστές. Η διαδικασία της κωδικοποίησης περιλαμβάνει συνδυασμό των αντίστοιχων macroblocks που παρουσιάζουν μικρές διαφορές με τα αντίστοιχα των καρέ αναφοράς (προηγούμενο και επόμενο) δηλαδή αφαίρεση του μέσου όρου των άλλων δύο από το τρέχον καρέ, συνδυασμό των διανυσμάτων κίνησης των καρέ αναφοράς (που συνδυάζονται όπως και τα αντίστοιχα macroblocks, δηλαδή λαμβάνεται ο μέσος όρος τους) και στη συνέχεια την ίδια διαδικασία με τα I και P καρέ για την κωδικοποίηση του macroblock που προκύπτει.

Οι φέτες SP και φέτες SI μπορούν να χρησιμοποιηθούν για την ανθεκτικότητα λάθους, για τη μετατροπή μεταξύ των διαφορετικών streams και για γρήγορη προώθηση. Οι φέτες SP χρησιμοποιούνται για να

κωδικοποιήσουν την ίδια φέτα χρησιμοποιώντας δύο διαφορετικά καρέ αναφοράς. Οι δύο φέτες SP αποκωδικοποιούνται ακριβώς στα ίδια στοιχεία. Μια φέτα SI κωδικοποιείται έτσι ώστε να αποκωδικοποιήσει ακριβώς τα ίδια στοιχεία με μια αντίστοιχη φέτα SP.

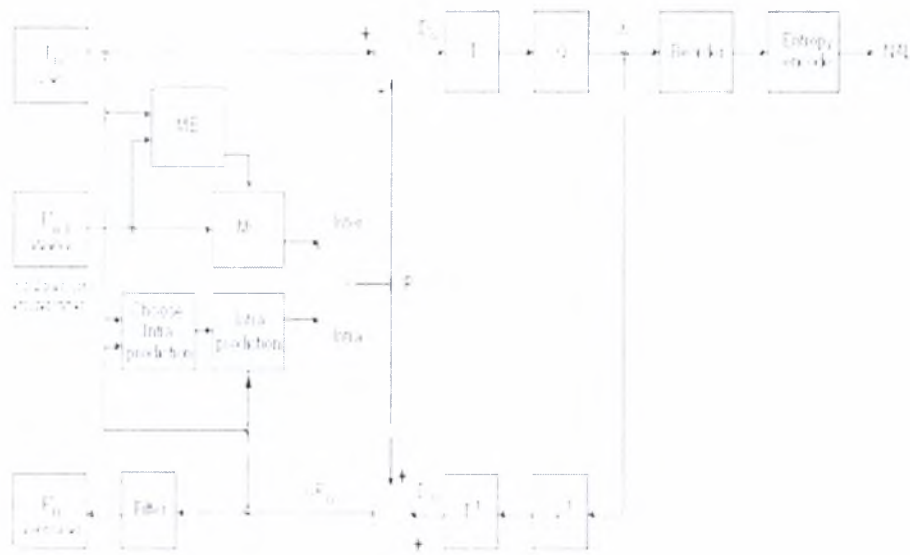
## 2. Η διαδικασία Κωδικοποίησης -Αποκωδικοποίησης

Ένα καρέ διαιρείται σε macroblocks (16x16 δειγμάτων). Τα macroblocks ομαδοποιούνται σε φέτες. Κάθε φέτα κωδικοποιείται είτε ως φέτα I, P είτε B (Υπάρχουν επίσης τύποι αποκαλούμενοι SI και SP). Σε μια φέτα I, όλα τα block κωδικοποιούνται ως block I. Σε μια φέτα P, τα block κωδικοποιούνται ως block I ή P. Σε μια φέτα B, τα block κωδικοποιούνται ως block I, P ή B.

Ο κωδικοποιητής περιλαμβάνει δύο κατευθύνσεις ροής πληροφοριών, μια αριστερά προς τα δεξιά (κωδικοποίηση) και μια δεξιά προς τα αριστερά (αναδημιουργία). Η κατεύθυνση ροής πληροφοριών στον αποκωδικοποιητή παρουσιάζεται από δεξιά στα αριστερά για να απεικονίσει τις ομοιότητες μεταξύ του κωδικοποιητή και του αποκωδικοποιητή. Πριν εξεταστεί λεπτομερώς το H.264, θα περιγράψουμε τα κύρια βήματα στην κωδικοποίηση και την αποκωδικοποίηση ενός καρέ βίντεο. Η ακόλουθη περιγραφή απλοποιείται προκειμένου να παρασχεθεί μια επισκόπηση των διαδικασιών κωδικοποίησης και αποκωδικοποίησης.

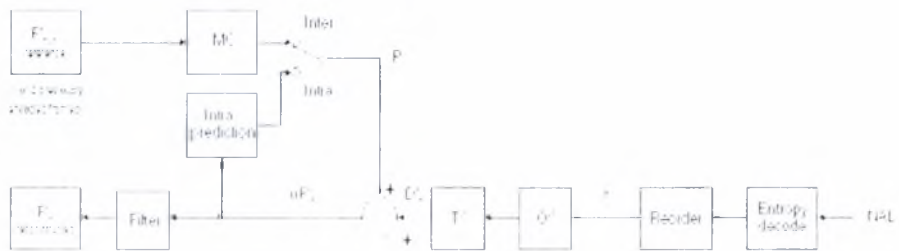
Ένα καρέ ή ένας τομέας εισαγωγής  $F_n$  υποβάλλεται σε επεξεργασία στις μονάδες ενός macroblock. Κάθε macroblock κωδικοποιείται με intra ή inter πρόβλεψη, και για κάθε μπλοκ στο macroblock, μια πρόβλεψη PRED (P μέσα στο  $F_n$ ) διαμορφώνεται βασισμένη στα αναδημιουργημένα δείγματα εικόνων. Στην intra πρόβλεψη, η PRED διαμορφώνεται από τα δείγματα στην τρέχουσα φέτα που έχουν κωδικοποιηθεί προηγουμένως, έχουν

αποκωδικοποιηθεί και έχουν αναδημιουργηθεί ( $uF'_n$  στο [εξότυπο 1.3](#), σημειώστε ότι τα μη φιλτραρισμένα δείγματα χρησιμοποιούνται για να διαμορφώσουν PRED). Στην inter πρόβλεψη, η PRED διαμορφώνεται από την motion-compensated πρόβλεψη από μια ή δύο εικόνες αναφοράς που επιλέγονται από το σύνολο εικόνων αναφοράς. Στις εικόνες, η εικόνα αναφοράς παρουσιάζεται ως προηγούμενη κωδικοποιημένη εικόνα  $F'_{n-1}$  αλλά η αναφορά πρόβλεψης για κάθε macroblock (στην inter πρόβλεψη) μπορεί να επιλεγεί από μια επιλογή των προηγούμενων ή μελλοντικών εικόνων που έχουν κωδικοποιηθεί ήδη, αναδημιουργηθεί και φιλτραριστεί. Η πρόβλεψη PRED αφαιρείται από τον τρέχον μπλοκ για να παραχθεί το μπλοκ υπολειπομένων (residual block)  $D_n$  που μετασχηματίζεται και κβαντοποιείται για να δώσει  $X$ , ένα σύνολο κβαντοποιημένων συντελεστών μετατροπής οι οποίοι έχουν αναδιαταχτεί και κωδικοποιηθεί. Οι κωδικοποιημένοι συντελεστές μέσω της εντροπίας, μαζί με την πληροφορία που απαιτείται για να αποκωδικοποιηθεί κάθε μπλοκ μέσα στο macroblock διαμορφώνουν το συμπιεσμένο bitstream το οποίο εισάγεται σε ένα στρώμα αφαίρεσης δικτύων (NAL) για μετάδοση ή αποθήκευση.



Ο κωδικοποιητής αναδημιουργεί κάθε μπλοκ για να παρέχει μια αναφορά για τις περαιτέρω προβλέψεις. Οι συντελεστές  $X$  κλιμακώνονται ( $Q^{-1}$ ) και μετασχηματίζονται ανίστροφα ( $T^{-1}$ ) για να παράγουν ένα υπολειπόμενο μπλοκ  $D'_n$  διαφοράς. Το μπλοκ πρόβλεψης PRED προστίθεται στο  $D'_n$  για να δημιουργηθεί ένα αναδημιουργημένο μπλοκ  $uF'_n$  (μια αποκωδικοποιημένη έκδοση του αρχικού μπλοκ, το  $u$  δείχνει ότι δεν είναι φιλτραρισμένο). Ένα φίλτρο εφαρμόζεται για να μειώσει τα αποτελέσματα της διαμόρφωσης του μπλοκ και η αναδημιουργημένη εικόνα αναφοράς δημιουργείται από μια σειρά μπλοκ  $F'_n$ .

Ο αποκωδικοποιητής λαμβάνει ένα συμπιεσμένο bitstream από το NAL και αποκωδικοποιεί τα δεδομένα μέσω εντροπίας για να παραγάγει ένα σύνολο κβαντοποιημένων συντελεστών  $X$ . Αυτοί κλιμακώνονται και μετασχηματίζονται ανίστροφα για να δώσουν  $D'_n$  (ίδιο με το  $D'_n$  που παρουσιάζεται στον κωδικοποιητή). Χρησιμοποιώντας τις πληροφορίες κεφαλίδας που αποκωδικοποιούνται από το bitstream, ο αποκωδικοποιητής δημιουργεί ένα μπλοκ πρόβλεψης PRED, ίδιο με τον αρχικό PRED που διαμορφώνεται στον κωδικοποιητή. Το τελευταίο προστίθεται στο  $D'_n$  για να παράγει  $uF'_n$  που φιλτράρονται για να δημιουργήσουν κάθε αποκωδικοποιημένο μπλοκ  $F'_n$ .



### 3. Το H.264 στην πράξη

Ίσως το μεγαλύτερο πλεονέκτημα του H.264 σε σχέση τα προηγούμενα πρότυπα είναι η απόδοση στην συμπίεση. Έναντι των προτύπων όπως Mpeg-2 και Mpeg-4, το H.264 μπορεί να επιτύχει:

- ❖ Καλύτερη ποιότητα εικόνας στο ίδιο συμπιεσμένο bitrate
- ❖ Ένα χαμηλότερο συμπιεσμένο bitrate για την ίδια ποιότητα εικόνας.

Παραδείγματος χάριν, ένα single-layer DVD μπορεί να αποθηκεύσει μια ταινία μήκους περίπου 2 ωρών σε MPEG-2 μορφή. Χρησιμοποιώντας το H.264, πρέπει να είναι δυνατό να αποθηκευτούν 4 ώρες ή περισσότερες της ταινίας σε ίδιο δίσκο (δηλ. χαμηλότερο bitrate). Εναλλακτικά, η μορφή συμπίεσης H.264 μπορεί να μεταδώσει καλύτερη ποιότητα στο ίδιο bitrate έναντι Mpeg-2 και Mpeg-4 οπτικά *perceptually*. Η βελτιωμένη απόδοση συμπίεσης H.264 έρχεται στην τιμή μεγαλύτερου υπολογιστικού κόστους. Το H.264 είναι περιπλοκότερο από τις προηγούμενες μεθόδους συμπίεσης και αυτό σημαίνει ότι μπορεί να πάρει σημαντικά περισσότερη ισχύς επεξεργασίας να συμπίεσει και αποσυμπίεσει το βίντεο H.264.



Παρόλο τη βελτιωμένη απόδοση συμπίεσής του, το H.264 προσφέρει μεγαλύτερη ευελιξία στις επιλογές συμπίεσης και της υποστήριξης μετάδοσης. Ένας κωδικοποιητής H.264 μπορεί να επιλέξει από μια ευρεία ποικιλία εργαλείων συμπίεσης, καθιστώντας τον κατάλληλο για εφαρμογές από το χαμηλός-ποσοστό δυαδικών ψηφίων και τη χαμηλή καθυστέρηση κινητής μετάδοσης. Τα πρότυπα παρέχουν ενσωματωμένη υποστήριξη για μετάδοση ή αποθήκευση, συμπεριλαμβανομένης μιας πακετοποιημένης συμπίεσμης μορφής και των χαρακτηριστικών γνωρισμάτων που βοηθούν για να ελαχιστοποιηθεί η επίδραση των λαθών μετάδοσης.

Το H.264/AVC υιοθετείται για μια αυξανόμενη σειρά εφαρμογών, που περιλαμβάνουν:

- ❖ Υψηλός καθορισμός DVD's (σχήματα hd-DVD και blu-ακτίνων)
- ❖ Υψηλή τηλεοπτική αναμετάδοση καθορισμού στην Ευρώπη
- ❖ Τα προϊόντα Apple
- ❖ Τηλεοπτικές εφαρμογές του NATO και των ΗΠΑ DoD
- ❖ Κινητή τηλεοπτική αναμετάδοση
- ❖ Βίντεο Διαδικτύου
- ❖ Συνεδρίαση μέσω βίντεο

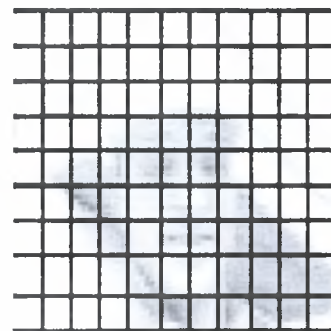


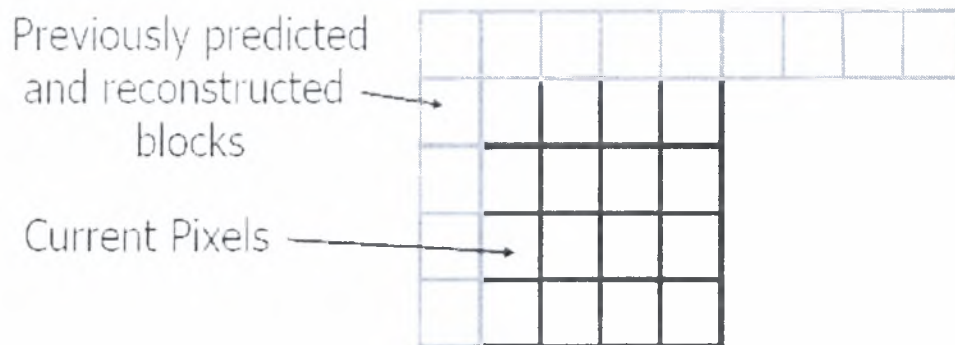
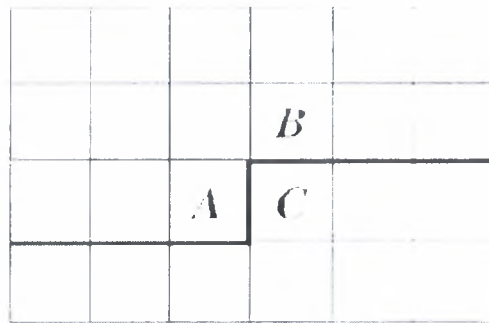
Το h.264 έχει κοινά σημεία με άλλα πρότυπα. Αυτά είναι

- ❖ 16x16 macroblocks
- ❖ Συμβατική 4 : 2 : 0 δειγματοληψία χρώματος και σύνδεση των στοιχείων φωτεινότητας και χρώματος
- ❖ Μετατόπιση κίνησης μπλοκ ( block motion displacement )
- ❖ Διανύσματα κίνησης πέρα από τα όρια εικόνων
- ❖ Μεταβλητή κίνηση μεγέθους μπλοκ
- ❖ Μετατροπές μπλοκ
- ❖ Κλιμακωτή κβαντοποίηση
- ❖ I, P και B τύποι εικόνας

## 1. Πρόβλεψη των intra macroblocks

Όπως έχουμε ήδη δει, ένα καρτέ εικόνας αρχικά χωρίζεται σε macroblocks (εικόνα 2.1). Κάθε macroblock αποτελείται από μικρότερα δομικά στοιχεία, τα οποία έχουμε ονομάσει μπλοκ (εικόνα 2.2). Ένα μπλοκ στην ουσία είναι ένας δισδιάστατος πίνακας και κάθε στοιχείο του είναι ένα pixel (εικόνα 2.3). Η τιμή κάθε pixel προκύπτει με την εφαρμογή των μεθόδων πρόβλεψης ενός macroblock. Υπάρχουν δυο μέθοδοι πρόβλεψης, η  $1D$  και η  $2D$ . Στην πρώτη, οι τιμές των pixels προκύπτουν από τιμές των pixels γειτονικών μπλοκ που βρίσκονται όμως **στο ίδιο καρτέ** ενώ στη δεύτερη προκύπτουν από τιμές των pixels μπλοκ που βρίσκονται σε **διαφορετικά καρτέ αλλά στην ίδια θέση με το τρέχον**. Για παράδειγμα, αν το τρέχον μπλοκ σε ένα καρτέ είναι το μπλοκ που βρίσκεται πάνω και αριστερά, οι τιμές των pixels του τρέχοντος μπλοκ θα προκύψουν από μπλοκ που βρίσκονται στην ίδια θέση με το τρέχον σε διαφορετικά καρτέ.





Πιο συγκεκριμένα στην intra πρόβλεψη, οι τιμές των pixels ανήκουν σε γειτονικά μπλοκ τα οποία έχουν κωδικοποιηθεί προηγουμένως και ξαναδημιουργηθεί αλλά δεν είναι φιλτραρισμένα (σχήμα 2.3). Αυτό το μπλοκ πρόβλεψης P αφαιρείται από το τρέχον μπλοκ πριν από την κωδικοποίηση. Για τα δείγματα φωτεινότητας (luma), το P μπορεί να διαμορφωθεί για κάθε 4x4 μπλοκ ή για ένα 16x16 macroblock. Υπάρχουν συνολικά

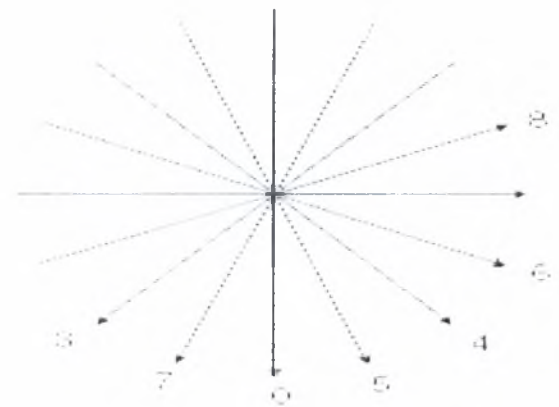
- ❖ 9 προαιρετικοί τρόποι πρόβλεψης για κάθε 4x4 luma μπλοκ (σχήμα 2.4)
- ❖ 4 προαιρετικοί τρόποι για έναν luma 16x16 μπλοκ
- ❖ και ένας τρόπος που εφαρμόζεται πάντα σε κάθε 4x4 μπλοκ χρώματος.

Μέθοδοι πρόβλεψης 4X4 luma μπλοκ.

Mode 0	Vertical
Mode 1	Horizontal
Mode 2	DC (πάντα πραγματοποιείται)
Mode 3	Diagonal down -left
Mode 4	Diagonal down -right
Mode 5	Vertical right
Mode 6	Horizontal-down
Mode 7	Vertical left
Mode 8	Horizontal-up

Μέθοδοι πρόβλεψης 16X16 luma μπλοκ.

Mode 0	Vertical
Mode 1	Horizontal
Mode 2	DC
Mode 3	Plane



## 2. Τρόποι πρόβλεψης 4X4 μπλοκ φωτεινότητας

Το [απόσπασμα 2.5](#) παρουσιάζει ένα luma macroblock και ένα 4x4 luma μπλοκ που πρέπει να προβλεφθεί. Τα δείγματα επάνω και αριστερά έχουν κωδικοποιηθεί προηγουμένως και αναδημιουργηθεί και είναι επομένως διαθέσιμα στον κωδικοποιητή και τον αποκωδικοποιητή για να διαμορφώσει μια αναφορά πρόβλεψης. Το μπλοκ πρόβλεψης P υπολογίζεται σύμφωνα με τα δείγματα A έως M στο [απόσπασμα 2.6](#), ως εξής. Η DC πρόβλεψη (τρόπος 0) τροποποιείται ανάλογα με το ποια δείγματα είναι διαθέσιμα. Οι άλλοι τρόποι (1-8) μπορούν μόνο χρησιμοποιηθούν μόνο εάν όλα τα απαραίτητα δείγματα πρόβλεψης είναι διαθέσιμα. Ο κωδικοποιητής μπορεί να επιλέξει τον τρόπο πρόβλεψης για κάθε μπλοκ που ελαχιστοποιεί το υπόλοιπο μεταξύ του P και του μπλοκ που κωδικοποιείται.



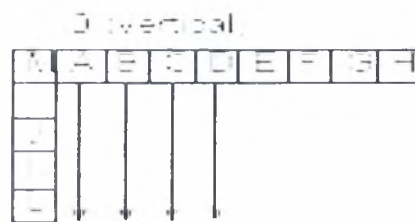
Original image (a)

4x4 Lyapunov image (b)

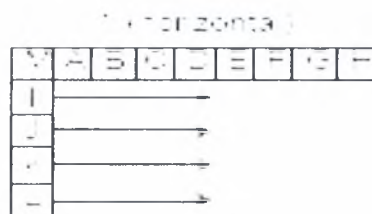


M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

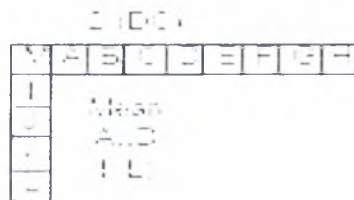
Τα δείγματα A,B,C,D που βρίσκονται στο πάνω μέρος του πίνακα προεκτείνονται κάθετα (vertical)



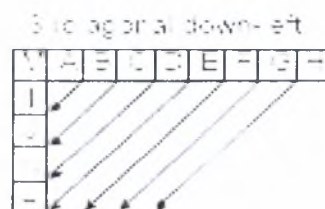
Τα δείγματα I,J,K,L που βρίσκονται αριστερά του πίνακα προεκτείνονται οριζόντια (horizontal)



Όλα τα δείγματα στο μπλοκ προβλέπονται από το μέσο όρο των δειγμάτων A,B,C,D και I,J,K,L



Τα δείγματα παρεμβάλλονται σε μια γωνία 45° μεταξύ του κάτω δεξιού δείγματος και άνω αριστερά





Τα δείγματα παρεκτείνονται σε μια γωνία  $45^\circ$  κάτω και προς τα αριστερά

(αριστερά)

4 το άξονα down-right

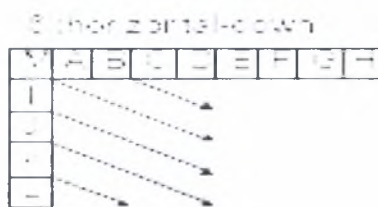


Παρέκταση διαγωνίως περίπου  $26.6^\circ$  αριστερά και κάθετα (πλάτος/ύψος =  $1/2$ ) (αριστερά)

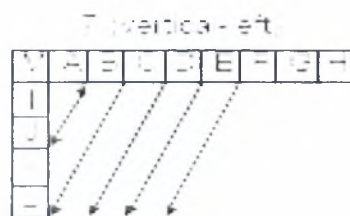
5 το άξονα down-right



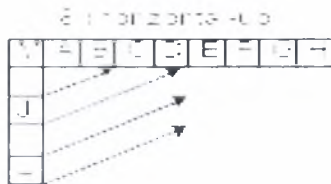
Παρέκταση διαγωνίως περίπου 26.6° οριζόντια και κάτω (πλάτος/ύψος = 1/2)



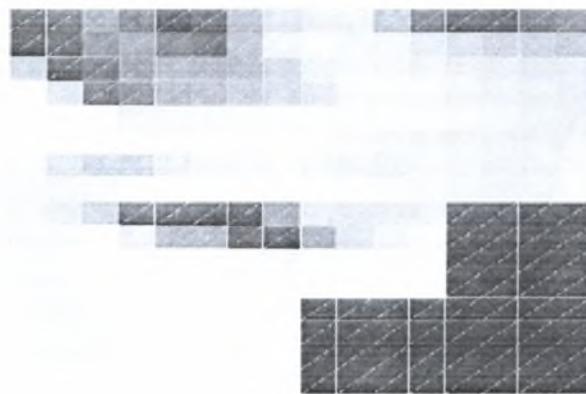
Προέκταση (ή παρεμβολή) διαγωνίως περίπου 26.6° δεξιά της κατακορύφου



Παρεμβολή διαγωνίως περίπου 26.6° επάνω και οριζόντια



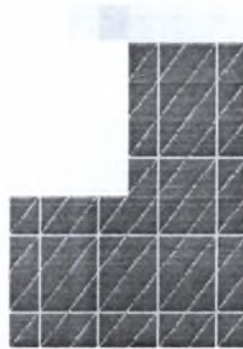
Οι εννέα τρόποι πρόβλεψης υπολογίζονται για την εικόνα του *εικονιστής 2.16* και τα μπλοκ πρόβλεψης  $P$  που προκύπτουν από κάθε τρόπο απεικονίζονται στην εικόνα του *εικονιστής 2.17*. Το ποσό των απόλυτων σφαλμάτων (SAE) για κάθε μία πρόβλεψη δείχνει το μέγεθος του λάθους πρόβλεψης. Σε αυτή την περίπτωση, το καλύτερο ταίριασμα στο *τρέχων καρέ* δίνεται από τον τρόπο 8 (horizontal-up) επειδή αυτός ο τρόπος δίνει το μικρότερο το SAE. Μια οπτική σύγκριση δείχνει ότι το μπλοκ  $P$  εμφανίζεται αρκετά παρόμοιο με τα αρχικό  $4 \times 4$  μπλοκ.



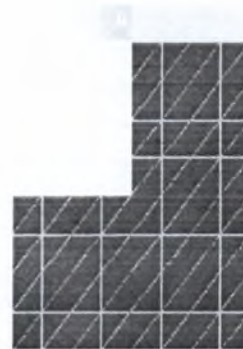
0 (vertical), SAE=619



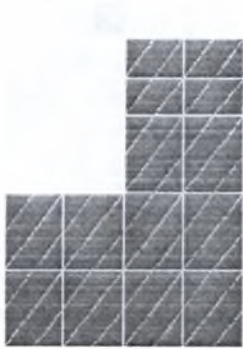
1 (horizontal), SAE=637



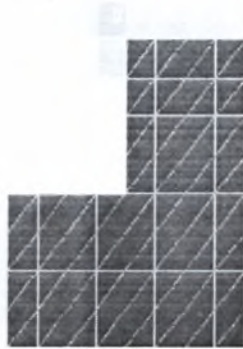
2 (0), SAE=607



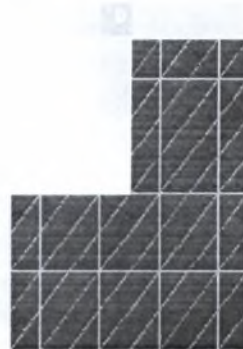
3 (rotat down-left), SAE=202



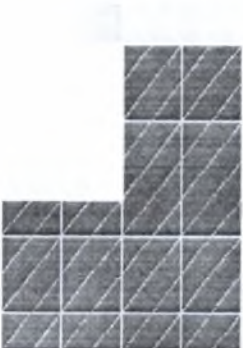
4 (diag down-right), SAE=1002



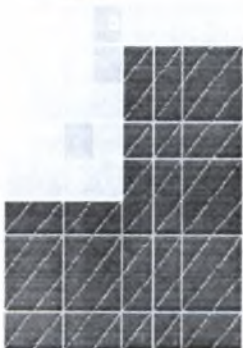
5 (vertical-right), SAE=308



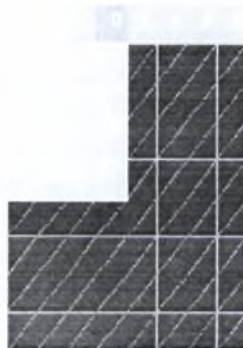
6 (horizontal-down), SAE=639



7 (vertical-left), SAE=167



8 (horizontal-up), SAE=333

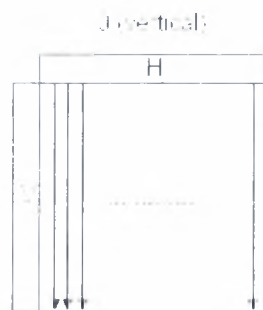


## 3. Τρόποι προβλεψής 16X16 μπλοκ φωτεινότητας

Σαν εναλλακτική λύση των τρόπων για τα 4X4 μπλοκ φωτεινότητας που περιγράφονται ανωτέρω, ένα ολόκληρο μπλοκ φωτεινότητας 16x16 μπορεί να προβλεφθεί με μια εκτέλεση. Τέσσερις τρόποι είναι διαθέσιμοι σε αυτή την περίπτωση

- ❖ Τρόπος 0 (κατακόρυφος)
- ❖ Τρόπος 1 (οριζόντιος)
- ❖ Τρόπος 2 (συνεχές ρεύμα): μέσος όρος ανώτερων και αριστερών δειγμάτων
- ❖ Τρόπος 4 (επίπεδο): μια γραμμική λειτουργία εγκαθίσταται στα ανώτερα και αριστερά δείγματα.

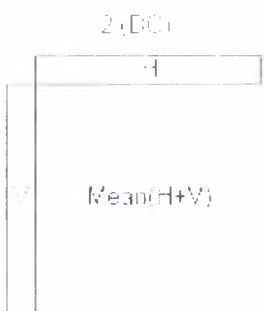
Τα δείγματα H που βρίσκονται στο πάνω μέρος του πίνακα προεκτείνονται κάθετα (εικόνα 2.18)



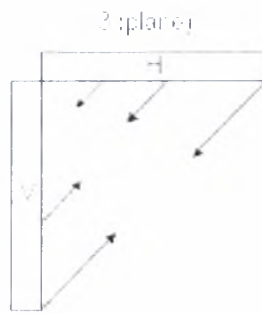
Τα δείγματα  $V$  που βρίσκονται αριστερά του πίνακα προεκτείνονται οριζόντια



Όλα τα δείγματα στο μπλοκ προβλέπονται από το μέσο όρο των δειγμάτων  $H$  και  $V$  (αρχικά  $\lambda, 20\%$ ).



Μια γραμμική επίπεδη συνάρτηση εγκαθίσταται στα ανώτερα και αριστερά δείγματα X και V.



Το [επιχείρημα 23](#) παρουσιάζει ένα μπλοκ φωτεινότητας με τα προηγούμενως κωδικοποιημένα δείγματα πάνω και αριστερά. Τα αποτελέσματα της πρόβλεψης [επιχείρημα 23](#) δείχνουν ότι η καλύτερη αντιστοιχία δίνεται από τον τρόπο 3.



2. Vertical, SAE#1090



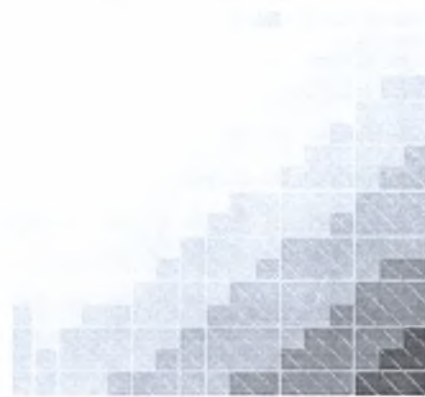
3. Horizontal, SAE#1090



4. Diagonal, SAE#11210



5. Diagonal, SAE#1214





#### 4 . Intra prediction 4X4 vs. Intra Prediction 16X16

Η πρόβλεψη για μπλοκ μεγέθους 4X4 είναι περισσότερη κατάλληλη για περιοχές που έχουν αρκετές λεπτομέρειες. Επίσης παρέχει περισσότερες επιλογές κατά τη διαδικασία πρόβλεψης ενός μπλοκ. Από τη άλλη η πρόβλεψη για μπλοκ μεγέθους 16X16 είναι κατάλληλη για περισσότερο ομαλές περιοχές. Η δουλεία του κωδικοποιητή είναι να συγκρίνει τα αποτελέσματα των διάφορων τρόπων και να επιλέξει την καλύτερη. Αυτό επιφέρει αρκετό κόστος.

#### 5 . Τρόποι πρόβλεψης 8X8 μπλοκ χρώματος

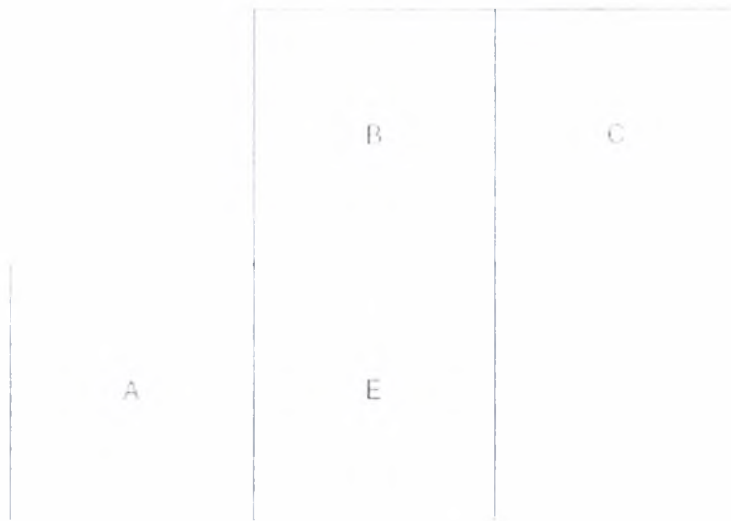
Κάθε 8x8 τμήμα χρώματος ενός μπλοκ (macroblock) προβλέπεται από τα δείγματα χρώματος επάνω και αριστερά που έχουν κωδικοποιηθεί προηγουμένως και έχει αναδημιουργηθεί. Οι 4 τρόποι πρόβλεψης είναι παρόμοιοι με τους τρόπους πρόβλεψης μπλοκ φωτεινότητας 16x16 που περιγράφονται στην παράγραφο 3 εκτός από το ότι η διάταξη των αριθμών τρόπου είναι διαφορετική. Δηλαδή

- ❖ Συνεχές ρεύμα (τρόπος 0),
- ❖ οριζόντιος (τρόπος 1)
- ❖ κατακόρυφος (τρόπος 2)
- ❖ Επίπεδη εφαρμογή συνάρτησης (τρόπος 3). Ο ίδιος τρόπος πρόβλεψης εφαρμόζεται πάντα και στους δύο μπλοκ χρώματος.

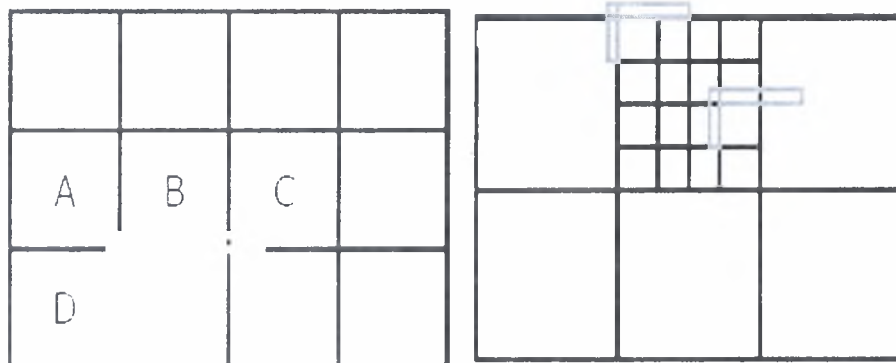
## 6. Επιλογή κατάλληλου τρόπου

Η επιλογή κάθε τρόπου πρόβλεψης για κάθε 4x4 μπλοκ πρέπει να επισημανθεί στον αποκωδικοποιητή και αυτό θα μπορούσε ενδεχομένως να απαιτεί έναν μεγάλο αριθμό από bits. Εντούτοις, οι τρόποι που τα γειτονικά 4x4 μπλοκ έχουν προβλεφθεί είναι ιδιαίτερα συσχετισμένοι. Παραδείγματος χάριν, ας θεωρήσουμε ότι τα μπλοκ A,B και E είναι το αριστερό μπλοκ, το πάνω μπλοκ και το τρέχον αντίστοιχα (παραδειγματικά). Εάν το A και το B προβλέφθηκαν χρησιμοποιώντας τον τρόπο 2, είναι πιθανό ότι ο καλύτερος τρόπος για το μπλοκ E (τρέχων μπλοκ) είναι επίσης ο τρόπος 2. Για κάθε τρέχοντα μπλοκ E, ο κωδικοποιητής και ο αποκωδικοποιητής υπολογίζουν την πιο πιθανή κατάσταση (most probable mode), δηλαδή την ελάχιστη από τις καταστάσεις πρόβλεψης του A και του B. Εάν κάποιο από το A ή το B δεν είναι διαθέσιμο (είτε γιατί δεν είναι κωδικοποιημένα με κάποιον τρόπο πρόβλεψης μπλοκ φωτεινότητας είτε γιατί δεν είναι στην τρέχουσα φέτα) τότε η τιμή της μεταβλητής most probable mode τίθεται 2 (συνεχές ρεύμα πρόβλεψη).

Ο κωδικοποιητής στέλνει μια σημαία για κάθε 4x4 καρέ, την `rem_intra4X4_pred_mode`. Εάν η σημαία είναι «1», η περισσότερο πιθανή κατάσταση πρόβλεψης χρησιμοποιείται. Εάν η σημαία είναι «0», μια άλλη παράμετρος η `rem_intra4X4_pred_mode` στέλνεται για να δείξει αλλαγή του τρόπου. Εάν η `rem_intra4X4_pred_mode` είναι μικρότερη από την τρέχουσα περισσότερο πιθανή κατάσταση πρόβλεψης, τότε ο τρόπος πρόβλεψης τίθεται στο `rem_intra4X4_pred_mode` διαφορετικά ο τρόπος πρόβλεψης τίθεται `rem_intra4X4_pred_mode + 1`.



Τέλος, όσον αφορά τα γειτονικά δείγματα, στις παρακάτω εικόνες **εικόνες** απεικονίζονται δυο περιπτώσεις στις οποίες φαίνεται ποια δείγματα από τα γειτονικά μπλοκ χρησιμοποιούνται. Με κόκκινο χρώμα απεικονίζονται τα διαθέσιμα δείγματα.



## 1. Εισαγωγή

Μέχρι τώρα έχουμε αναλύσει τα χαρακτηριστικά του προτύπου H.264 όπως επίσης και τις βασικές αρχές λειτουργίας του. Ιδιαίτερη έμφαση έχει δοθεί στην διαδικασία πρόβλεψης intra και συγκεκριμένα στους εννέα διαφορετικούς τρόπους πρόβλεψης για τα μπλοκ φωτεινότητας μεγέθους 4X4. Δηλαδή

Mode 0	Vertical
Mode 1	Horizontal
Mode 2	DC
Mode 3	Diagonal down -left
Mode 4	Diagonal down -right
Mode 5	Vertical right
Mode 6	Horizontal-down
Mode 7	Vertical left
Mode 8	Horizontal-up

Επίσης αναφέραμε και τους 4 τρόπους πρόβλεψης για τα μπλοκ φωτεινότητας μεγέθους 16X16. Δηλαδή

Mode 0	Vertical
Mode 1	Horizontal
Mode 2	DC
Mode 3	Plane

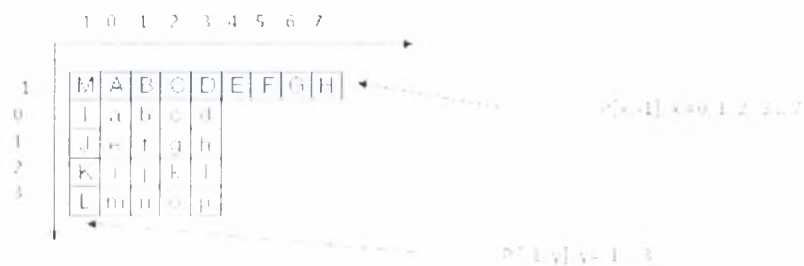
Στην συνέχεια αυτού του κεφαλαίου θα αναλύσουμε πως αυτοί οι τρόποι – συναρτήσεις περιγράφονται σε μαθηματική μορφή και θα εντυπώσουμε σε θέματα υλοποίησης.

## 2.1 Σχεδίαση τρόπων πρόβλεψης 4X4 μπλοκ φωτεινότητας.

Όπως έχουμε αναφέρει και σε προηγούμενο κεφάλαιο, κατά την διαδικασία της intra πρόβλεψης για ένα macroblock ενός καρτέ βίντεο, πριν την κωδικοποίηση, υπολογίζονται και οι εννέα τρόποι πρόβλεψης ενός μπλοκ και στην συνέχεια επιλέγεται ο καλύτερος τρόπος από τους εννέα. Ως καλύτερος τρόπος χαρακτηρίζεται αυτός που έχει τον μικρότερο ποσό απόλυτων σφαλμάτων. Έπειτα το μπλοκ πρόβλεψης που προκύπτει αφαιρείται από το τρέχον μπλοκ και σχηματίζεται το μπλοκ υπολοίπων (residual block) το οποίο και κωδικοποιείται. Για να είναι εφικτή μια μέθοδος πρόβλεψης, πρέπει ορισμένα ή και όλα τα δείγματα από τα γειτονικά μπλοκ που βρίσκονται πέριξ του τρέχοντος μπλοκ να είναι διαθέσιμα. Το ποια δείγματα πρέπει να είναι διαθέσιμα εξαρτάται από την συνάρτηση – τρόπο πρόβλεψης. Για παράδειγμα, ο κάθετος τρόπος πρόβλεψης ενός μπλοκ 4X4 (vertical) απαιτεί να είναι διαθέσιμα τα δείγματα που βρίσκονται άνω του τρέχοντος μπλοκ.

Σύμφωνα με το πρότυπο H.264, τα 13 γειτονικά δείγματα τα οποία είναι ήδη κατασκευασμένα πριν τη διαδικασία του deblocking filter τοποθετούνται σε ένα πίνακα  $p[x, y]$  με  $0 \leq x < 13$  και  $0 \leq y < 4$ .

- ❖  $x = -1, y = -1..3$  (δείγματα M, I, J, K, L –σε πορτοκαλί φόντο)
- ❖  $x = 0..7, y = -1$  (δείγματα A, B, C, D, E, F, G, H –σε κόκκινο φόντο )



Δηλαδή το δείγμα M βρίσκεται στη θέση  $p[-1,-1]$ , το δείγμα A στη θέση  $p[0,-1]$  και το δείγμα I βρίσκεται στη θέση  $p[-1,0]$ . Τα αποτελέσματα από την εφαρμογή κάθε τρόπου πρόβλεψης θα αποθηκεύονται κάθε φορά σε έναν δυοδιάστατο πίνακα  $pred4x4[x, y]$ , με  $x, y = 0..3$ .

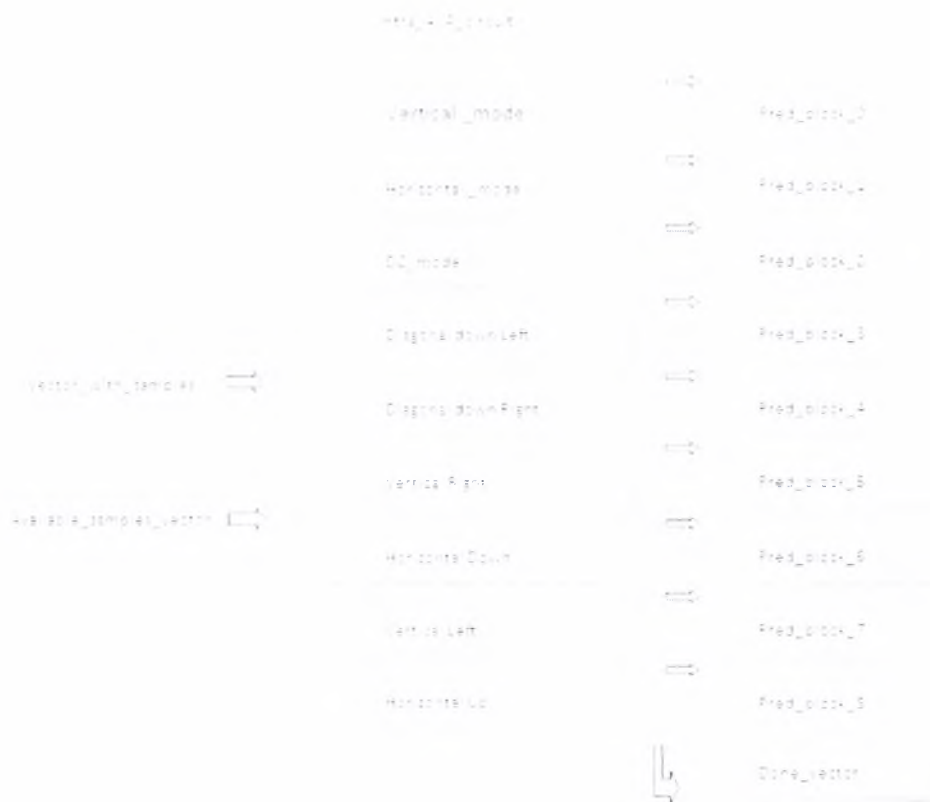
Σε μια γλώσσα περιγραφής υλικού όπως είναι η VHDL, ένα κύκλωμα που θα πραγματοποιούσε ολόκληρη την διαδικασία την πρόβλεψης θα ήταν το κύκλωμα το οποίο θα δέχονταν ως **είσοδο**

- ❖ ένα διάνυσμα με τα ήδη κατασκευασμένα δείγματα των γειτονικών μπλοκ του τρέχοντος πλαισίου
- ❖ ένα διάνυσμα ίδιου μεγέθους με το διάνυσμα των δειγμάτων το οποίο θα υποδηλώνει ποια δείγματα είναι διαθέσιμα

και θα παράγει ως **έξοδο**

- ❖ τα εννέα διαφορετικά μπλοκ από τους εννέα διαφορετικούς τρόπους πρόβλεψης
- ❖ ένα διάνυσμα μεγέθους εννέα θέσεων το οποίο θα υποδηλώνει ποια συνάρτηση - τρόπος έχει υλοποιηθεί

Το block diagram του κυκλώματος που περιγράψαμε ποιο πάνω είναι το ακόλουθο



Όσον αφορά τους τύπους και τα περιεχόμενα των διανυσμάτων εισόδου και εξόδου πρέπει να αναφέρουμε τα εξής στοιχεία:

1. το διάνυσμα *Vector\_with\_samples* περιέχει τα δείγματα των γειτονικών μπλοκ του τρέχοντος πλαισίου. Τα δείγματα είναι τύπου *byte*.

2. το διάνυσμα *Available\_samples\_vector* έχει το ίδιο μέγεθος με το διάνυσμα *Vector with samples*. Σε κάθε του θέση μπορεί να πάρει είτε '1' είτε '0'. Το '1' στην θέση *j* του πίνακα δείχνει ότι το δείγμα *j* είναι διαθέσιμο ενώ το '0' στη ίδια θέση δείχνει ότι το δείγμα δεν είναι διαθέσιμο. Για παράδειγμα αν το *Available\_samples\_vector[2]='1'*, αυτό σημαίνει ότι το 3<sup>ο</sup> δείγμα είναι διαθέσιμο για πρόβλεψη.
3. Τα διανύσματα *Pred\_block\_X* με  $X=0,\dots,8$  είναι ένας δυοδιάστατος πίνακας μεγέθους 4X4. Κάθε θέση του συγκεκριμένου πίνακα θα περιέχει την τιμή του συγκεκριμένου δείγματος. Τα δείγματα είναι τύπου byte.
4. Το διάνυσμα *Done\_vector* έχει μέγεθος 9 –όσοι δηλαδή και οι τρόποι πρόβλεψης ενός 4X4 μπλοκ φωτεινότητας. Σε κάθε του θέση μπορεί να πάρει είτε '1' είτε '0'. Το '1' στην θέση *j* του πίνακα υποδηλώνει ότι ο τρόπος πρόβλεψης με κωδικό αριθμό *j* έχει υλοποιηθεί ενώ το '0' ότι δεν έχει υλοποιηθεί. Για παράδειγμα εάν το *Done\_Vector[2]='1'*, αυτό σημαίνει ότι η DC συνάρτηση έχει υλοποιηθεί.



Έστω ότι τα δείγματα των γειτονικών μπλοκ του τρέχοντος μπλοκ είναι τα ακόλουθα που φαίνονται στο παρακάτω σχήμα με ετικέτες A,B,C,D,E,F,G,H,I,J,K,L,M. Τα δείγματα αυτά θα τοποθετηθούν στο διάνυσμα *Vector with samples* με την εξής σειρά :

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

M	A	B	C	D	E	F	G	H	M	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---	---	---

Δηλαδή ,στην θέση [0] του πίνακα βρίσκεται το δείγμα M ,στην θέση 1 το δείγμα A. Παρατηρούμε ότι το δείγμα M βρίσκεται σε 2 θέσεις του πίνακα. Αυτό γίνεται για ευκολία στην υλοποίηση.

**Για λόγους ευκολίας το διάνυσμα *Vector\_with\_samples* θα το συμβολίζουμε *s* στην συνέχεια.**

Τα δείγματα A,B,C,D που βρίσκονται στο πάνω μέρος του πίνακα προεκτείνονται κάθετα προς τα κάτω. Για να υλοποιηθεί αυτός ο τρόπος πρέπει αυτά τα δείγματα να είναι διαθέσιμα (τα δείγματα σε κίτρινο φόντο).



Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

$$\text{pred4x4}[x, y] = p[x, -1], \text{ με } x, y = 0 \dots 3$$

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται σε:

$$\text{Pred\_block}[0][x, y] = s[1] \text{ με } x, y = 0 \dots 3 \ \& \ 1 \dots 4$$

Τα δείγματα I,J,K,L που βρίσκονται αριστερά του πίνακα προεκτείνονται οριζόντια (αριστερά-δεξιά). Για να υλοποιηθεί αυτός ο τρόπος πρέπει αυτά τα δείγματα να είναι διαθέσιμα (τα δείγματα σε κίτρινο φόντο).



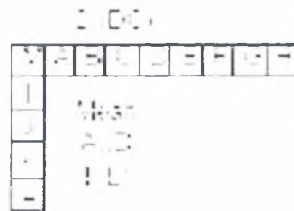
Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

$$\text{pred}_{4 \times 4}[x, y] = p[x-1, y], \text{ με } x, y = 0 \dots 3$$

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται σε :

$$\text{Pred\_block\_1}[x, y] = s(J) \text{ με } x, y = 0 \dots 3 \ \& \ J = 11, 12, 13, 14$$

Όλα τα δείγματα στο μπλοκ προβλέπονται από το μέσο όρο των δειγμάτων A,B,C,D και I,J,K,L. Για να υλοποιηθεί αυτός ο τρόπος πρέπει αυτά τα δείγματα να είναι διαθέσιμα (τα δείγματα σε κίτρινο φόντο).



Αυτός ο τρόπος πρόβλεψης πραγματοποιείται πάντα. Είναι ο μοναδικός που μπορεί να το πετύχει αυτό. Ο τρόπος αυτός έχει τέσσερις υποπεριπτώσεις.

Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

$$\text{pred4x4}[x, y] = (p[-1, 0] + p[-1, 1] + p[-1, 2] + p[-1, 3] + p[0, -1] + p[0, 0] + p[0, 1] + p[0, 2] + p[0, 3] + p[1, -1] + p[1, 0] + p[1, 1] + p[1, 2] + p[1, 3] + p[2, -1] + p[2, 0] + p[2, 1] + p[2, 2] + p[2, 3] + p[3, -1] + p[3, 0] + p[3, 1] + p[3, 2] + p[3, 3]) >> 3$$

με  $x, y = 0 \dots 3$

Εάν είναι διαθέσιμα μόνο τα δείγματα που βρίσκονται αριστερά τότε η μαθηματική αναπαράσταση της συνάρτησης είναι :

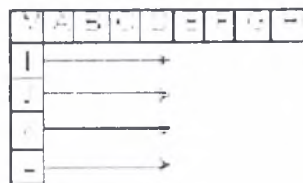
$$\text{pred4x4}_l[x, y] = (p[-1, 0] + p[-1, 1] + p[-1, 2] + p[-1, 3] + 2) >> 2$$

με  $x, y = 0 \dots 3$

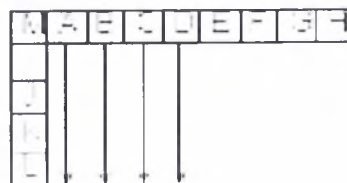
Εάν είναι διαθέσιμα μόνο τα δείγματα που βρίσκονται πάνω (αρχήμα 3.7 β), τότε η μαθηματική αναπαράσταση της συνάρτησης είναι :

$$\text{pred4x4}_i | x, y | = ( p[ 0, -1 ] + p[ 1, -1 ] + p[ 2, -1 ] + p[ 3, -1 ] + 2 ) \gg 2$$

με  $x, y = 0 \dots 3$



(α)



(β)

Εάν ένα είτε κανένα από τα δείγματα που βρίσκονται πάνω και αριστερά δεν είναι διαθέσιμα, τότε η μαθηματική αναπαράσταση της συνάρτησης είναι η ακόλουθη :

$$\text{pred4x4}_i | x, y | = 128$$

με  $x, y = 0 \dots 3$

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται σε:

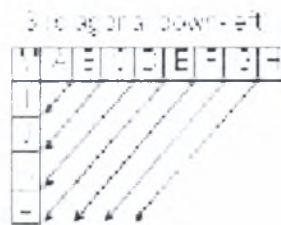
```
Pred_block_2[x,y] = (Σi s(i) + Σi s(J) + 4) >> 3  
με x, y = 0..3 & I = 1,2,3,4 & J = 11, 12, 13, 14
```

```
Pred_block_2[x,y] = (Σi s(J) + 2) >> 2  
με x, y = 0..3 & J = 11, 12, 13, 14
```

```
Pred_block_2[x,y] = (Σi s(i) + 2) >> 2  
με x, y = 0..3 & I = 1,2,3,4
```

```
Pred_block_2[x,y] = 128  
με x, y = 0..3
```

Τα δείγματα παρεμβάλλονται σε μια γωνία 45° μεταξύ του κάτω δεξιά δείγματος και άνω αριστερά (αρχικά 3mm). Για να υλοποιηθεί αυτός ο τρόπος πρέπει τα δείγματα A,B,C,D & E,F,G,H να είναι διαθέσιμα (τα δείγματα σε κίτρινο φόντο).



Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

$$\text{pred4x4}[x, y] = (\text{p}[6, -1] + 3 * \text{p}[7, -1] + 2) >> 2$$

$$\text{pred4x4}[x, y] = (\text{p}[x + y, -1] + 2 * \text{p}[x + y + 1, -1] + \text{p}[x + y + 2, -1] + 2) >> 2$$

με  $x, y = 0 \dots 3$

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται σε:

```
Pred block 3|x,y|=( s (7) +3 * s (8) +2 )>>2
```

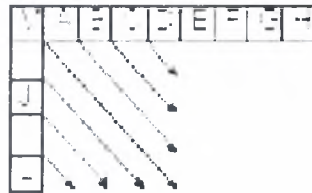
```
Pred block 3|x,y|=( s (x+y+1) +2 * s (x+y+2) + s (x+y+3) +2 )>>2
```

με  $x, y = 0..3$



Τα δείγματα προεκτείνονται σε μια γωνία 45° κάτω και προς τα αριστερά (εικόνα 10.10). Για να υλοποιηθεί αυτός ο τρόπος πρέπει τα δείγματα A,B,C,D & I,J,k,L & M να είναι διαθέσιμα (τα δείγματα σε κίτρινο φόντο).

Εικόνα 10.10: Δείγματα προέκτασης



Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

$$\text{pred4x4}_i[x, y] = (p[x - y - 2, -1] + 2 * p[x - y - 1, -1] + p[x - y, -1] + 2) >> 2$$

$$\text{pred4x4}_i[x, y] = (p[-1, y - x - 2] + 2 * p[-1, y - x - 1] + p[-1, y - x] + 2) >> 2$$

$$\text{pred4x4}_i[x, y] = (p[0, -1] + 2 * p[-1, -1] + p[-1, 0] + 2) >> 2$$

με  $x, y = 0 \dots 3$

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται με αντικατάσταση σε:

για τις θέσεις (1,0) ,(2,1) ,(3,2)  $\text{Pred\_block } 4|x,y| = (s(0) + 2 * s(1) + s(2) + 2) \gg 2$

για τις θέσεις (2,0) ,(3,1)  $\text{Pred\_block } 4|x,y| = (s(1) + 2 * s(2) + s(3) + 2) \gg 2$

για τη θέση (3,0)  $\text{Pred\_block } 4|x,y| = (s(2) + 2 * s(3) + s(4) + 2) \gg 2$

για τις θέσεις (0,1) ,(1,2) ,(2,3)  $\text{Pred\_block } 4|x,y| = (s(9) + 2 * s(10) + s(11) + 2) \gg 2$

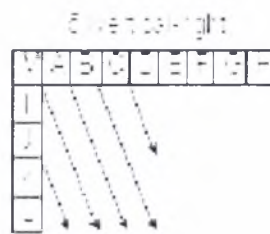
για τις θέσεις (0,2) ,(1,3)  $\text{Pred\_block } 4|x,y| = (s(10) + 2 * s(11) + s(12) + 2) \gg 2$

για τη θέση (0,3)  $\text{Pred\_block } 4|x,y| = (s(11) + 2 * s(12) + s(13) + 2) \gg 2$

για τις θέσεις (0,0) ,(1,1) ,(2,2) ,(3,3)  $\text{Pred\_block } 4|x,y| = (s(1) + 2 * s(0) + s(10) + 2) \gg 2$

με  $x, y \in \{0, \dots, 3\}$

Προέκταση διαγωνίως περίπου 26.6° αριστερά και κάθετα (πλάτος/ύψος = 1/2) [1]. Για να υλοποιηθεί αυτός ο τρόπος πρέπει τα δείγματα A,B,C,D & I,J,k,L & M να είναι διαθέσιμα (τα δείγματα σε κίτρινο φόντο).



Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

Έστω η μεταβλητή  $zVR = (2 * x) - y$

$$pred4x4_i[x, y] = (p[x - (y >> 1) - 1, -1] + p[x - (y >> 1), -1] + 1) >> 1$$

$$pred4x4_i[x, y] = (p[x - (y >> 1) - 2, -1] + 2 * p[x - (y >> 1) - 1, -1] + p[x - (y >> 1), -1] + 2) >> 2$$

$$pred4x4_i[x, y] = (p[-1, 0] + 2 * p[-1, -1] + p[0, -1] + 2) >> 2$$

$$pred4x4_i[x, y] = (p[-1, y - 1] + 2 * p[-1, y - 2] + p[-1, y - 3] + 2) >> 2$$

με  $x, y = 0...3$

Στους παραπάνω τύπους παρατηρούμε ότι το  $y$  υπόκειται σε δεξιά ολίσθηση ενός ψηφίου.

Εάν δώσουμε τιμές στις μεταβλητές  $x, y = 0..3$  τότε

Στην περίπτωση όπου το  $zVR$  ισούται με  $0,2,4,6$  παρατηρούμε ότι θα επηρεάζονται οι θέσεις πίνακα που φαίνονται στον παραπάνω πίνακα. Σε αυτές τις περιπτώσεις το  $y$  ισούται με  $0$  ή  $2$ . Εάν κάνουμε δεξιά ολίσθηση στο  $0$  παίρνουμε ξανά  $0$ . Εάν κάνουμε στο  $2$ , παίρνουμε  $1$ .

Στην περίπτωση όπου το  $zVR$  ισούται με  $1,3,5$  παρατηρούμε ότι θα επηρεάζονται οι θέσεις πίνακα που φαίνονται στον παραπάνω πίνακα. Σε αυτές τις περιπτώσεις το  $y$  ισούται με  $1$  ή  $3$ . Εάν κάνουμε δεξιά ολίσθηση στο  $1$  παίρνουμε ξανά  $0$ . Εάν κάνουμε στο  $3$ , παίρνουμε  $1$ . Συνεπώς οι τύποι του προτύπου απλοποιούνται ως εξής :

Έτσι η μεταβλητή zVR = (2 \* x) - y

pred4x4[ x, y ] = ( p[ x - 1, - 1 ] + p[ x, - 1 ] + 1 ) >> 1

pred4x4[ x, y ] = ( p[ x - 2, - 1 ] + p[ x - 1, - 1 ] + 1 ) >> 1

pred4x4[ x, y ] = ( p[ x - 2, - 1 ] + 2 \* p[ x - 1, - 1 ] + p[ x, - 1 ] + 2 ) >> 2

pred4x4[ x, y ] = ( p[ x - 3, - 1 ] + 2 \* p[ x - 2, - 1 ] + p[ x - 1, - 1 ] + 2 ) >> 2

pred4x4[ x, y ] = ( p[ - 1, 0 ] + 2 \* p[ - 1, - 1 ] + p[ 0, - 1 ] + 2 ) >> 2

pred4x4[ x, y ] = ( p[ - 1, y - 1 ] + 2 \* p[ - 1, y - 2 ] + p[ - 1, y - 3 ] + 2 ) >> 2

με x, y = 0...3

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται με αντικατάσταση σε

Έτσι η μεταβλητή  $zVR = (2 * x) - y$

για τις θέσεις (0,0) ,(1,2)      Pred block  $S[x,y] = (s(0) + s(1) + 1) >> 1$

για τις θέσεις (1,0) ,(2,2)      Pred block  $S[x,y] = (s(1) + s(2) + 1) >> 1$

για τις θέσεις (2,0) ,(3,2)      Pred block  $S[x,y] = (s(2) + s(3) + 1) >> 1$

για τις θέσεις (3,0)              Pred block  $S[x,y] = (s(3) + s(4) + 1) >> 1$

για τις θέσεις (1,1) ,(2,3)      Pred block  $S[x,y] = (s(0) + 2 * s(1) + s(2) + 2) >> 2$

για τις θέσεις (2,1) ,(3,3)      Pred block  $S[x,y] = (s(1) + 2 * s(2) + s(3) + 2) >> 2$

για τις θέσεις (3,1)              Pred block  $S[x,y] = (s(2) + 2 * s(3) + s(4) + 2) >> 2$

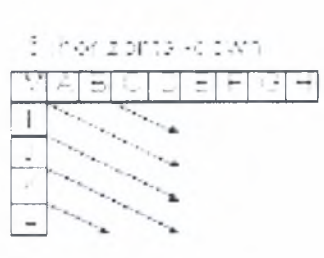
για τις θέσεις (0,1) ,(1,3)      Pred block  $S[x,y] = (s(10) + 2 * s(9) + s(1) + 2) >> 2$

για τις θέσεις (0,2)              Pred block  $S[x,y] = (s(11) + 2 * s(10) + s(9) + 2) >> 2$

για τις θέσεις (0,3)              Pred block  $S[x,y] = (s(12) + 2 * s(11) + s(10) + 2) >> 2$

με  $x, y = 0 \dots 3$

Προέκταση διαγωνίως περίπου 26.6° οριζόντια και κάτω (πλάτος/ύψος = 1/2) (από το κέντρο). Για να υλοποιηθεί αυτός ο τρόπος πρέπει τα δείγματα A,B,C,D & I,J,k,L & M να είναι διαθέσιμα (τα διαθέσιμα δείγματα σε κίτρινο φόντο).



Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

```

// στω η μεταβλητή zHD = (2 * γ) - x
pred4x4[| x, y |] = ( p[ -1, y - ( x >> 1 ) - 1 ] + p[ -1, y - ( x >> 1 ) + 1 ] ) >> 1

pred4x4[| x, y |] = ( p[ -1, y - ( x >> 1 ) - 2 ] + 2 * p[ -1, y - ( x >> 1 ) - 1 ] +
                    p[ -1, y - ( x >> 1 ) + 2 ] ) >> 2

pred4x4[| x, y |] = ( p[ -1, 0 ] + 2 * p[ -1, -1 ] + p[ 0, -1 ] + 2 ) >> 2

pred4x4[| x, y |] = ( p[ x - 1, -1 ] + 2 * p[ x - 2, -1 ] + p[ x - 3, -1 ] + 2 ) >> 2
με x, y = 0...3
    
```

Στους παραπάνω τύπους παρατηρούμε ότι το  $x$  υπόκειται σε δεξιά ολίσθηση ενός ψηφίου.

Εάν δώσουμε τιμές στις μεταβλητές  $x, y = 0 \dots 3$  τότε

Στην περίπτωση όπου το  $z_{HD}$  ισούται με  $0, 2, 4, 6$  παρατηρούμε ότι θα επηρεάζονται οι θέσεις πίνακα που φαίνονται στον παραπάνω πίνακα. Σε αυτές τις περιπτώσεις το  $x$  ισούται με  $0$  ή  $2$ . Εάν κάνουμε δεξιά ολίσθηση στο  $0$  παίρνουμε ξανά  $0$ . Εάν κάνουμε στο  $2$ , παίρνουμε  $1$ .

Στην περίπτωση όπου το  $z_{HD}$  ισούται με  $1, 3, 5$  παρατηρούμε ότι θα επηρεάζονται οι θέσεις πίνακα που φαίνονται στον παραπάνω πίνακα. Σε αυτές τις περιπτώσεις το  $x$  ισούται με  $1$  ή  $3$ . Εάν κάνουμε δεξιά ολίσθηση στο  $1$  παίρνουμε ξανά  $0$ . Εάν κάνουμε στο  $3$ , παίρνουμε  $1$ . Συνεπώς οι τύποι του προτύπου απλοποιούνται ως εξής :



Έστω η μεταβλητή zHD = (2 \* y) - x

pred4x4<sub>i</sub>[ x, y ] = ( p[ -1, y - 1 ] + p[ -1, y ] + 1 ) >> 1

pred4x4<sub>i</sub>[ x, y ] = ( p[ -1, y - 2 ] + p[ -1, y - 1 ] + 1 ) >> 1

pred4x4<sub>i</sub>[ x, y ] = ( p[ -1, y - 2 ] + 2 \* p[ -1, y - 1 ] + p[ -1, y ] + 2 ) >> 2

pred4x4<sub>i</sub>[ x, y ] = ( p[ -1, y - 3 ] + 2 \* p[ -1, y - 2 ] + p[ -1, y - 1 ] + 2 ) >> 2

pred4x4<sub>i</sub>[ x, y ] = ( p[ -1, 0 ] + 2 \* p[ -1, - 1 ] + p[ 0, - 1 ] + 2 ) >> 2

pred4x4<sub>i</sub>[ x, y ] = ( p[ -1, y - 1 ] + 2 \* p[ -1, y - 2 ] + p[ -1, y - 3 ] + 2 ) >> 2

με x, y = 0...3

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται με αντικατάσταση σε

Έτσι η μεταβλητή  $zHD = (2 * y) - x$

για τις θέσεις (0,0) ,(2,1)      Pred block  $6[x,y] = (s(9) + s(10) + 1) >> 1$

για τις θέσεις (0,1) ,(2,2)      Pred block  $6[x,y] = (s(10) + s(11) + 1) >> 1$

για τις θέσεις (0,2) ,(2,3)      Pred block  $6[x,y] = (s(11) + s(123) + 1) >> 1$

για τις θέσεις (0,3)              Pred block  $6[x,y] = (s(12) + s(13) + 1) >> 1$

για τις θέσεις (1,1) ,(3,2)      Pred block  $6[x,y] = (s(9) + 2 * s(10) + s(11) + 2) >> 2$

για τις θέσεις (1,2) ,(3,3)      Pred block  $6[x,y] = (s(10) + 2 * s(11) + s(12) + 2) >> 2$

για τις θέσεις (1,3)              Pred block  $6[x,y] = (s(11) + 2 * s(12) + s(13) + 2) >> 2$

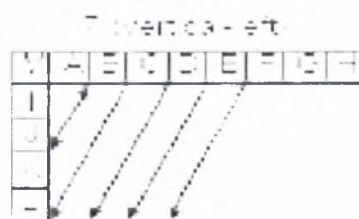
για τις θέσεις (1,0) ,(3,1)      Pred block  $6[x,y] = (s(10) + 2 * s(11) + s(1) + 2) >> 2$

για τις θέσεις (2,0)              Pred block  $6[x,y] = (s(2) + 2 * s(1) + s(0) + 2) >> 2$

για τις θέσεις (3,0)              Pred block  $6[x,y] = (s(3) + 2 * s(2) + s(1) + 2) >> 2$

με  $x, y = 0...3$

Προέκταση (ή παρεμβολή) διαγωνίως περίπου 26.6° δεξιά της κατακόρυφου  $\theta = 0^\circ$ . Για να υλοποιηθεί αυτός ο τρόπος πρέπει τα δείγματα A,B,C,D & E,F,G,H να είναι διαθέσιμα (τα διαθέσιμα δείγματα σε κίτρινο φόντο).



Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

$$\text{pred4x4}_i[x, y] = (p[x + (y \gg 1), -1] + p[x + (y \gg 1) + 1, -1] + 1) \gg 1$$

$$\text{pred4x4}_i[x, y] = (p[x + (y \gg 1), -1] + 2 * p[x + (y \gg 1) + 1, -1] + p[x + (y \gg 1) + 2, -1] + 2) \gg 2$$

με  $x, y = 0 \dots 3$

Στους παραπάνω τύπους παρατηρούμε ότι το  $y$  υπόκειται σε δεξιά ολίσθηση ενός ψηφίου .

Εάν δώσουμε τιμές στις μεταβλητές  $x, y = 0 \dots 3$  τότε

Στην περίπτωση όπου το  $y$  ισούται με 0,2 παρατηρούμε ότι θα επηρεάζονται οι θέσεις πίνακα που φαίνονται στον παραπάνω πίνακα. Σε αυτές τις περιπτώσεις το  $y$  ισούται με 0 ή 2. Εάν κάνουμε δεξιά ολίσθηση στο 0 παίρνουμε ξανά 0. Εάν κάνουμε στο 2, παίρνουμε 1.

Στην περίπτωση όπου το  $y$  ισούται με 1,3 παρατηρούμε ότι θα επηρεάζονται οι θέσεις πίνακα που φαίνονται στον παραπάνω πίνακα. Σε αυτές τις περιπτώσεις το  $y$  ισούται με 1 ή 3. Εάν κάνουμε δεξιά ολίσθηση στο 1 παίρνουμε ξανά 0. Εάν κάνουμε στο 3, παίρνουμε 1. Συνειπώς οι τύποι του προτύπου απλοποιούνται ως εξής :

$$\text{pred4x4}_i[x, y] = (p[x, -1] + p[x + 1, -1] + 1) \gg 1$$

$$\text{pred4x4}_i[x, y] = (p[x + 1, -1] + p[x + 2, -1] + 1) \gg 1$$

$$\text{pred4x4}_i[x, y] = (p[x, -1] + 2 * p[x + 1, -1] + p[x + 2, -1] + 2) \gg 2$$

$$\text{pred4x4}_i[x, y] = (p[x + 1, -1] + 2 * p[x + 2, -1] + p[x + 3, -1] + 2) \gg 2$$

με  $x, y = 0 \dots 3$

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται με αντικατάσταση σε

για τις θέσεις (0,0)                      Pred block  $7[x,y] = (s(1) + s(2) + 1) \gg 1$

για τις θέσεις (1,0) ,(0,2)              Pred block  $7[x,y] = (s(2) + s(3) + 1) \gg 1$

για τις θέσεις (2,0) ,(1,2)              Pred block  $7[x,y] = (s(3) + s(4) + 1) \gg 1$

για τις θέσεις (3,0),(2,2)              Pred block  $7[x,y] = (s(4) + s(5) + 1) \gg 1$

για τις θέσεις (3,2)                      Pred block  $7[x,y] = (s(5) + s(6) + 1) \gg 1$

για τις θέσεις (0,1)                      Pred block  $7[x,y] = (s(1) + 2 * s(2) + s(3) + 2) \gg 2$

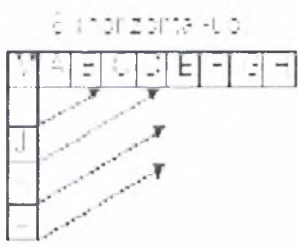
για τις θέσεις (1,1) ,(0,3)              Pred block  $7[x,y] = (s(2) + 2 * s(3) + s(4) + 2) \gg 2$

για τις θέσεις (1,3),(2,1)              Pred block  $7[x,y] = (s(3) + 2 * s(4) + s(5) + 2) \gg 2$

για τις θέσεις (3,1),(2,3)              Pred block  $7[x,y] = (s(4) + 2 * s(5) + s(6) + 2) \gg 2$

για τις θέσεις (3,3)                      Pred block  $7[x,y] = (s(5) + 2 * s(6) + s(7) + 2) \gg 2$

Παρεμβολή διαγωνίως περίπου 26.6° επάνω και οριζόντια (αριθμός 3, 3, 3). Για να υλοποιηθεί αυτός ο τρόπος πρέπει αυτά τα δείγματα να είναι διαθέσιμα (τα διαθέσιμα δείγματα σε κίτρινο φόντο).



Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

Έστω η μεταβλητή  $zHU = x + 2 * y$ .

$$pred4x4_i[x, y] = (p[-1, y + (x >> 1)] + p[-1, y + (x >> 1) + 1] + 1) >> 1$$

$$pred4x4_i[x, y] = (p[-1, y + (x >> 1)] + 2 * p[-1, y + (x >> 1) + 1] + p[-1, y + (x >> 1) + 2] + 2) >> 2$$

$$pred4x4_i[x, y] = (p[-1, 2] + 3 * p[-1, 3] + 2) >> 2$$

$$pred4x4_i[x, y] = p[-1, 3]$$

Στους παραπάνω τύπους παρατηρούμε ότι το  $x$  υπόκειται σε δεξιά ολίσθηση ενός ψηφίου .

Εάν δώσουμε τιμές στις μεταβλητές  $x, y = 0..3$  τότε

Στην περίπτωση όπου το  $zHU$  ισούται με  $0,2,4$  παρατηρούμε ότι θα επηρεάζονται οι θέσεις πίνακα που φαίνονται στον παραπάνω πίνακα. Σε αυτές τις περιπτώσεις το  $x$  ισούται με  $0$  ή  $2$ . Εάν κάνουμε δεξιά ολίσθηση στο  $0$  παίρνουμε ξανά  $0$ . Εάν κάνουμε στο  $2$ , παίρνουμε  $1$ .

Στην περίπτωση όπου το  $zHD$  ισούται με  $1,3$  παρατηρούμε ότι θα επηρεάζονται οι θέσεις πίνακα που φαίνονται στον παραπάνω πίνακα. Σε αυτές τις περιπτώσεις το  $x$  ισούται με  $1$  ή  $3$ . Εάν κάνουμε δεξιά ολίσθηση στο  $1$  παίρνουμε ξανά  $0$ . Εάν κάνουμε στο  $3$ , παίρνουμε  $1$ . Συνεπώς οι τύποι του προτύπου απλοποιούνται ως εξής:

Έτσι η μεταβλητή  $zHU = x + 2 * y$ .

```
pred4x4[ x, y ] = ( p[ -1, y ] + p[ -1, y + 1 ] + 1 ) >> 1
```

```
pred4x4[ x, y ] = ( p[ -1, y + 1 ] + p[ -1, y + 2 ] + 1 ) >> 1
```

```
pred4x4[ x, y ] = ( p[ -1, y ] + 2 * p[ -1, y + 1 ] + p[ -1, y + 2 ] + 2 ) >> 2
```

```
pred4x4[ x, y ] = ( p[ -1, y + 1 ] + 2 * p[ -1, y + 2 ] + p[ -1, y + 3 ] + 2 ) >> 2
```

```
pred4x4[ x, y ] = ( p[ -1, 2 ] + 3 * p[ -1, 3 ] + 2 ) >> 2
```

```
pred4x4[ x, y ] = p[ -1, 3 ]
```



Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται με αντικατάσταση σε

Έστω η μεταβλητή  $z_{i,j} = (2 * y) - x$

για τις θέσεις (0,0)  $\text{Pred\_block } 8|x,y| = (s(10) + s(11) + 1) \gg 1$

για τις θέσεις (0,1),(2,0)  $\text{Pred\_block } 8|x,y| = (s(11) + s(12) + 1) \gg 1$

για τις θέσεις (0,2),(2,1)  $\text{Pred\_block } 8|x,y| = (s(12) + s(13) + 1) \gg 1$

για τις θέσεις (1,0)  $\text{Pred\_block } 8|x,y| = (s(10) + 2 * s(11) + s(12) + 2) \gg 2$

για τις θέσεις (1,1),(3,0)  $\text{Pred\_block } 8|x,y| = (s(11) + 2 * s(12) + s(13) + 2) \gg 2$

για τις θέσεις (1,2),(3,1)  $\text{Pred\_block } 8|x,y| = (s(12) + 3 * s(13) + 2) \gg 2$

για τις θέσεις (0,3), (1,3), (2,2), (2,3), (3,2), (3,3)  $\text{Pred\_block } 8|x,y| = s(13)$

### 3. Σχεδίαση τρόπων πρόβλεψης 16X16 μπλοκ φωτεινότητας.

Η διαδικασία κωδικοποίησης και αποκωδικοποίησης που ακολουθείται για τα μπλοκ φωτεινότητας 16X16 είναι ίδια με αυτή των μπλοκ 4X4. Σύμφωνα με το *πρότυπο H.264*, τα 33 γειτονικά δείγματα τα οποία είναι ήδη κατασκευασμένα πριν τη διαδικασία του deblocking filter τοποθετούνται σε ένα πίνακα  $p[x, y]$  με

$$\clubsuit x = -1, y = -1..15$$

$$\clubsuit x = 0..15, y = -1$$

Σε μία γλώσσα περιγραφής υλικού όπως είναι η VHDL, ένα κύκλωμα που θα πραγματοποιούσε ολόκληρη την διαδικασία την πρόβλεψης θα ήταν κύκλωμα το οποίο θα δέχονταν ως **είσοδο**

- ❖ ένα διάνυσμα με τα ήδη κατασκευασμένα δείγματα των γειτονικών μπλοκ του τρέχοντος πλαισίου
- ❖ ένα διάνυσμα ίδιου μεγέθους με το διάνυσμα των δειγμάτων το οποίο θα υποδηλώνει ποια δείγματα είναι διαθέσιμα

και θα παράγει ως **έξοδο**

- ❖ τα τέσσερα διαφορετικά μπλοκ από τους τέσσερις διαφορετικούς τρόπους πρόβλεψης
- ❖ ένα διάνυσμα μεγέθους τεσσάρων θέσεων το οποίο θα υποδηλώνει ποια συνάρτηση – τρόπος έχει υλοποιηθεί

Το block diagram του κυκλώματος που περιγράψαμε ποιο πάνω είναι το ακόλουθο:



Όσον αφορά τους τύπους και τα περιεχόμενα των διανυσμάτων εισόδου και εξόδου πρέπει να αναφέρουμε τα εξής στοιχεία:

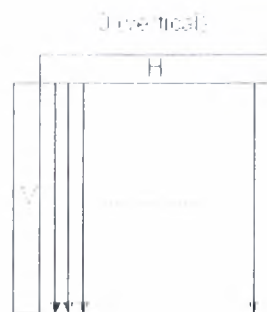
1. το διάνυσμα *Vector\_with\_samples* περιέχει τα δείγματα των γειτονικών μπλοκ του τρέχοντος πλαισίου. Τα δείγματα είναι τύπου `byte`.
2. το διάνυσμα *Available\_samples\_vector* έχει το ίδιο μέγεθος με το διάνυσμα *Vector\_with\_samples*. Σε κάθε του θέση μπορεί να πάρει είτε '1' είτε '0'. Το '1' στην θέση *j* του πίνακα δείχνει ότι το δείγμα *j* είναι διαθέσιμο ενώ το '0' στη ίδια θέση δείχνει ότι το δείγμα δεν είναι διαθέσιμο. Για παράδειγμα αν το *Available\_samples\_vector[2]='1'*, αυτό σημαίνει ότι το 3<sup>ο</sup> δείγμα είναι διαθέσιμο για πρόβλεψη.
3. Τα διανύσματα *Pred\_block\_X* με  $X=0\dots3$  είναι ένας δυοδιάστατος πίνακας μεγέθους 16X16. Κάθε θέση του συγκεκριμένου πίνακα θα περιέχει την τιμή του συγκεκριμένου δείγματος. Τα δείγματα είναι τύπου `byte`.
4. Το διάνυσμα *Done\_vector* έχει μέγεθος 4 –όσοι δηλαδή και οι τρόποι πρόβλεψης ενός 16X16 μπλοκ φωτεινότητας. Σε κάθε του θέση μπορεί να πάρει είτε '1' είτε '0'. Το '1' στην θέση *j* του πίνακα υποδηλώνει ότι ο τρόπος πρόβλεψης με κωδικό αριθμό *j* έχει

υλοποιηθεί ενώ το '0' ότι δεν έχει υλοποιηθεί. Για παράδειγμα εάν το Done Vector[2]='1', αυτό σημαίνει ότι η DC συνάρτηση έχει υλοποιηθεί .

Έστω H και V τα δείγματα που βρίσκονται πάνω και αριστερά αντίστοιχα από το τρέχον μπλοκ 16X16. Στο διάνυσμα *Vector\_with\_samples* θα τοποθετηθεί στο στη θέση 1 το δείγμα  $p[-1,-1]$ , στις θέσεις 1 έως 16 τα δείγματα H, στη θέση 17 το δείγμα  $p[-1,-1]$  για λόγους ευκολίας στην υλοποίηση και στις θέσεις 18 έως 33 τα δείγματα V.

**Για λόγους ευκολίας το διάνυσμα *Vector\_with\_samples* θα το συμβολίζουμε *s* στην συνέχεια**

Τα δείγματα  $H$  που βρίσκονται στο πάνω μέρος του πίνακα προεκτείνονται κάθετα  $\uparrow$ . Για να υλοποιηθεί αυτός ο τρόπος πρέπει τα δείγματα να είναι διαθέσιμα (τα δείγματα σε κίτρινο φόντο).



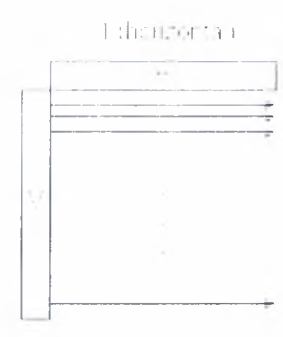
Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

$$\text{Pred16X16}[x, y] = p[x, -1], \text{ με } x, y = 0 \dots 15$$

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται σε :

$$\text{Pred\_block}_0[x, y] = s[-1] \text{ με } x, y = 0 \dots 15 \ \& \ I = 1 \dots 16$$

Τα δείγματα  $V$  που βρίσκονται αριστερά του πίνακα προεκτείνονται οριζόντια (από  $x=0$  έως  $x=15$ ). Για να υλοποιηθεί αυτός ο τρόπος πρέπει τα δείγματα να είναι διαθέσιμα (τα δείγματα σε κίτρινο φόντο).



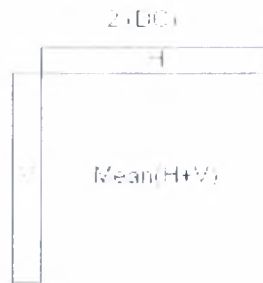
Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

$$\text{Pred}_{16 \times 16}[x, y] = p[x-1, y], \text{ με } x, y = 0 \dots 15$$

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται σε

$$\text{Pred\_block\_1}[x, y] = s(J), \text{ με } x, y = 0 \dots 15 \text{ \& } J = 18 \dots 33$$

Όλα τα δείγματα στο μπλοκ προβλέπονται από το μέσο όρο των δειγμάτων H και V (  $\frac{H+V}{2}$  ). Για να υλοποιηθεί αυτός ο τρόπος πρέπει αυτά τα δείγματα να είναι διαθέσιμα (τα διαθέσιμα δείγματα σε κίτρινο φόντο).



Αυτός ο τρόπος πρόβλεψης πραγματοποιείται πάντα. Είναι ο μοναδικός που μπορεί να το πετύχει αυτό. Ο τρόπος αυτός έχει τέσσερις υποπεριπτώσεις.

Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

$$\text{Pred16X16}[x, y] = (\sum_x p[x', -1] + \sum_y p[-1, y'] + 16) \gg 5$$

με  $x, y = 0 \dots 15$  &  $x', y' = 0 \dots 15$

Εάν είναι διαθέσιμα μόνο τα δείγματα που βρίσκονται αριστερά (  $\sum_x p[x', -1]$  ), τότε η μαθηματική αναπαράσταση της συνάρτησης είναι :

$$\text{Pred16X16}[x, y] = (\sum_x p[x', -1] + 8) \gg 4$$

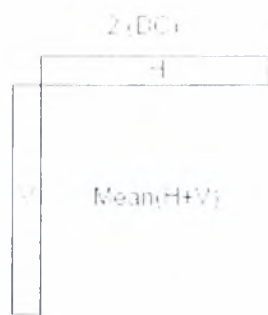
με  $x, y = 0 \dots 15$  &  $y' = 0 \dots 15$

Εάν είναι διαθέσιμα μόνο τα δείγματα που βρίσκονται πάνω

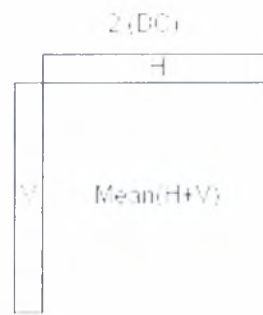
τότε η μαθηματική αναπαράσταση της συνάρτησης είναι :

$$\text{Pred}_{16 \times 16}[x, y] = (\sum_{x'} p[x', 1] + 8) \gg 4$$

$$\text{με } x, y = 0 \dots 15 \quad \& \quad x' = 0 \dots 15$$



(α)



(β)

Εάν ένα είτε κανένα από τα δείγματα που βρίσκονται πάνω και αριστερά δεν είναι διαθέσιμα, τότε η μαθηματική αναπαράσταση της συνάρτησης είναι η ακόλουθη :

$$\text{Pred}_{16 \times 16}[x, y] = 128$$

$$\text{με } x, y = 0 \dots 3$$



Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται σε:

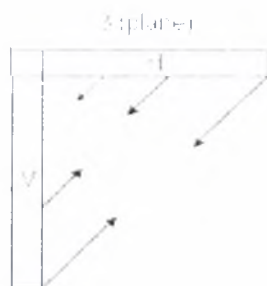
```
Pred_block 2[x,y] = (Σi s(i) + Σj s(j)+16) >> 5  
με x, y = 0..15 & I = 1..16 & J = 17..33
```

```
Pred_block 2[x,y] = (Σj s(j)+8) >> 4  
με x, y = 0..15 & J = 17..33
```

```
Pred_block 2[x,y] = (Σi s(i) + 8) >> 4  
με x, y = 0..15 & I = 1..16
```

```
Pred_block 2[x,y] = 128  
με x, y = 0..15
```

Επίπεδη γραμμική συνάρτηση (σχήμα 3.18). Για να υλοποιηθεί αυτός ο τρόπος πρέπει όλα τα δείγματα να είναι διαθέσιμα (τα δείγματα σε κίτρινο φόντο).



Η μαθηματική αναπαράσταση της παραπάνω συνάρτησης σύμφωνα με το πρότυπο είναι:

$$\text{Pred16X16}[x, y] = \text{Clip1}((a + b * (x-7) + c * (y-7) + 16) >> 5)$$

$$\text{Clip1}(x) = \text{Clip3}(0, 255, x)$$

$$\text{Clip3}(x, y, z) = \begin{cases} x, & z < x \\ y, & z > y \\ z, & \text{αλλιώς} \end{cases}$$

$$a = 16 * (p[-1, 15] + p[15, -1])$$

$$b = (5 * H + 32) >> 6$$

$$c = (5 * V + 32) >> 6$$

$$H = \sum_x (x'+1) * (p[8+x',-1] - p[6-x',-1])$$

$$V = \sum_y (y'+1) * (p[-1, 8+y'] - p[-1, 6+y'])$$

$$\text{με } x, y = 0 \dots 15 \quad x', y' = 0 \dots 7$$

Για τις ανάγκες της υλοποίησης η συνάρτηση μετατρέπεται σε:

$$\text{Pred\_block } 3[x,y] = \text{Clip1}((a + b * (x / 7) + c * (y / 7) + 16) >> 5)$$

$$\text{Clip1}(x) = \text{Clip3}(0, 255, x)$$

$$\text{Clip3}(x) = \begin{cases} 0, & \text{εάν } x < 0 \\ 255, & \text{εάν } x > 255 \\ x, & \text{αλλιώς} \end{cases}$$

$$a = 16 * (s(33) + s(16))$$

$$b = (5 * H + 32) >> 6$$

$$c = (5 * V + 32) >> 6$$

$$H = (\sum_x (x' + 1) * (s(9 + x') - s(7 - x')))$$

$$V = (\sum_y (y' + 1) * (s(26 + y') - s(24 - y')))$$

$$\text{με } x, y = 0 \dots 15 \quad x', y' = 0 \dots 7$$

## 1 . Εισαγωγή

Στο προηγούμενο κεφάλαιο, περιγράψαμε την μαθηματική αναπαράσταση των συναρτήσεων - τρόπων πρόβλεψης για τα μπλοκ φωτεινότητας μεγέθους 4X4 και 16X16 δειγμάτων. Πιο συγκεκριμένα αναφέραμε την μαθηματική αναπαράσταση των συναρτήσεων σύμφωνα με το πρότυπο H.264. Στη συνέχεια επισημάναμε πως οι συναρτήσεις μετατρέπονται για τις ανάγκες τις υλοποίησης.

## 2 . Υλοποίηση μεθόδων πρόβλεψης για τα μπλοκ φωτεινότητας 4X4

Εάν μελετήσουμε καλύτερα τις τροποποιημένες μαθηματικές αναπαραστάσεις των συναρτήσεων - τρόπων πρόβλεψης, θα παρατηρήσουμε ότι πολλές εκφράσεις οι οποίες αναθέτουν τιμή στα στοιχεία ενός μπλοκ εκτός από εκείνες για την dc πρόβλεψη είναι κοινές σε περισσότερους του ενός τρόπων πρόβλεψης. Οι διαφορετικές εκφράσεις είναι οι ακόλουθες .

1.  $(s(0) +s(1)+1) >>1$

2.  $(s(1) +s(2)+1) >>1$

3.  $(s(2) + s(3) + 1) \gg 1$

4.  $(s(3) + s(4) + 1) \gg 1$

5.  $(s(4) + s(5) + 1) \gg 1$

6.  $(s(5) + s(6) + 1) \gg 1$

7.  $(s(7) + 3 * s(8) + 2) \gg 2$

8.  $(s(12) + 3 * s(13) + 2) \gg 2$

9.  $(s(0) + 2 * s(1) + s(2) + 2) \gg 2$

10.  $(s(1) + 2 * s(2) + s(3) + 2) \gg 2$

$$11.(s(2) + 2 * s(3) + s(4) + 2) >> 2$$

$$12.(s(3) + 2 * s(4) + s(5) + 2) >> 2$$

$$13.(s(4) + 2 * s(5) + s(6) + 2) >> 2$$

$$14.(s(5) + 2 * s(6) + s(7) + 2) >> 2$$

$$15.(s(6) + 2 * s(7) + s(8) + 2) >> 2$$

$$16.(s(9) + 2 * s(10) + s(11) + 2) >> 2$$

$$17.(s(10) + 2 * s(11) + s(12) + 2) >> 2$$

$$18.(s(11) + 2 * s(12) + s(13) + 2) >> 2$$

$$19.(s(1) + 2 * s(0) + s(10) + 2) >> 2$$

$$20.(s(10) + 2 * s(9) + s(1) + 2) >> 2$$

$$21.(s(11) + 2 * s(10) + s(9) + 2) >> 2$$

$$22.(s(12) + 2 * s(11) + s(10) + 2) >> 2$$

$$23.(s(10) + 2 * s(11) + s(1) + 2) >> 2$$

$$24.(s(2) + 2 * s(1) + s(0) + 2) >> 2$$

$$25.(s(3) + 2 * s(2) + s(1) + 2) >> 2$$

$$26.(s(9) + s(10) + 1) >> 1$$

$$27.(s(10) + s(11)+1)>>1$$

$$28.(s(11) + s(12)+1)>>1$$

$$29.(s(12)+s(13) +1)>>1$$

Τα στοιχεία του διανύσματος  $s()$  αναπαριστούν τις τιμές των γειτονικών δειγμάτων. Το πεδίο τιμών των δειγμάτων είναι από 0...255 στο δεκαδικό σύστημα. Επομένως στο δυαδικό σύστημα οι τιμές των δειγμάτων θα αποτελούνται από 8 bits ( $2^8 = 255$  διαφορετικοί συνδυασμοί). Από τα 29 παραπάνω αθροίσματα παρατηρούμε ότι υπάρχουν 3 διαφορετικά είδη αθροισμάτων

- $(s(i) + s(j) + 1) >> 1$
- $(s(i) + 2 * s(j) + s(k) + 2) >> 2$
- $(s(i) + 3 * s(j) + 2) >> 2$

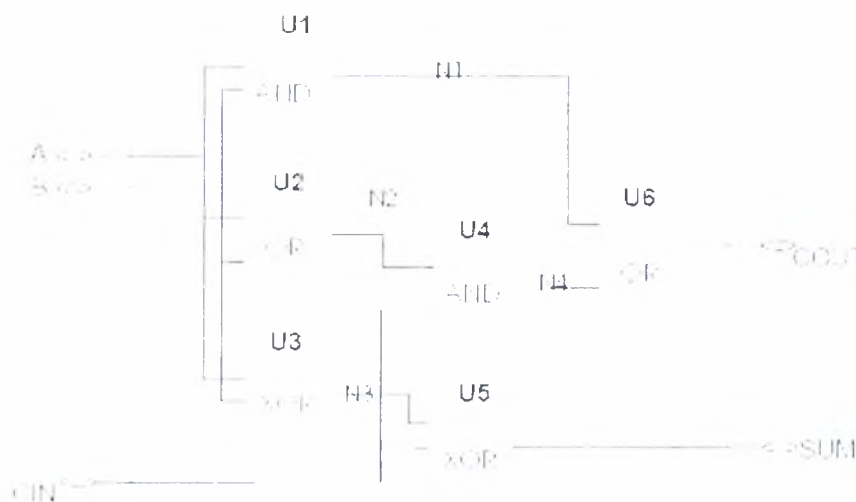
*Παρατηρήσεις:*

- Για τις παραπάνω προσθέσεις θα χρησιμοποιηθούν σαν δομικά στοιχεία αθροιστές 8 ,9 ,10 και 11 bits. Ο λόγος για τον οποίο χρησιμοποιούνται αθροιστές 9 ,10 ,11 bits ενώ τα δείγματα είναι 8 bits είναι για να μην χάνονται τα κρατούμενα (cout) και παραποιείται το αποτέλεσμα. Ο τελεστής & ενώνει το cout με το sum κάθε αθροιστή. Για παράδειγμα εάν cout='1' και sum='00010110' θα



δημιουργηθεί το 9bit σήμα cout & sum = '100010110' το οποίο θα αποτελέσει είσοδο σε έναν 9bit αθροιστή.

- Σαν δομικό στοιχείο των παραπάνω αθροιστών χρησιμοποιείται ο παρακάτω αθροιστής 1 bit ο οποίος απεικονίζεται στο [αρχείο 9\\_1\\_1](#).



- Ο τελεστής >> συμβολίζει δεξιά ολίσθηση τόσο ψηφία όσα και ο αριθμός που ακολουθεί τον τελεστή. Για παράδειγμα, >>2 στο 10 θα τον μετατρέψει σε 00
- Η πράξη  $k * s(i)$  μπορεί να υλοποιηθεί ως εξής :
  1. Εάν το  $k=2^j$ ,  $j=1\dots$  τότε η παραπάνω πράξη μπορεί να υπολογιστεί με αριστερή ολίσθηση  $j$  ψηφίων. Για παράδειγμα,  $2 * s(1)$ , με  $s(1) = 0$  θα μετατραπεί σε 0.
  2. Αλλιώς με πολλαπλασιασμό χρησιμοποιώντας σαν δομικό μπλοκ ένα πολλαπλασιαστή 8 bits είτε με διαδοχικές προσθέσεις χρησιμοποιώντας αθροιστές. Για παράδειγμα η

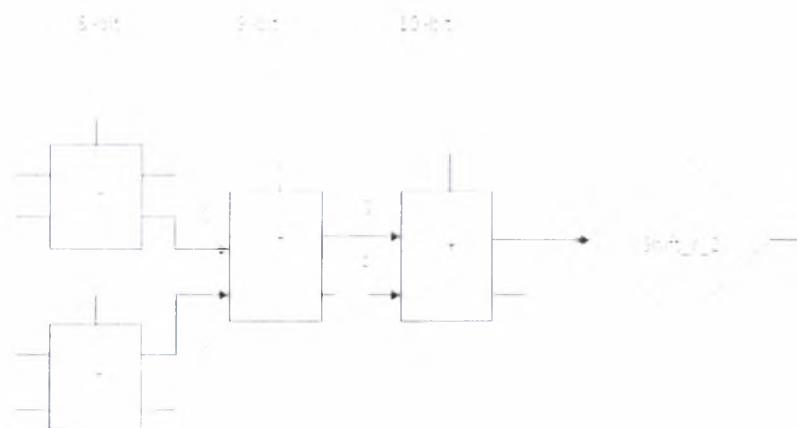
η πράξη  $3 * s(i)$  θα γίνει είτε πολλαπλασιάζοντας το  $s(i)$  με το 3  
είτε προσθέτοντας το  $s(i)$  3 φορές.

Τα κυκλώματα που πραγματοποιούν τις παραπάνω προσθέσεις είναι τα ακόλουθα:

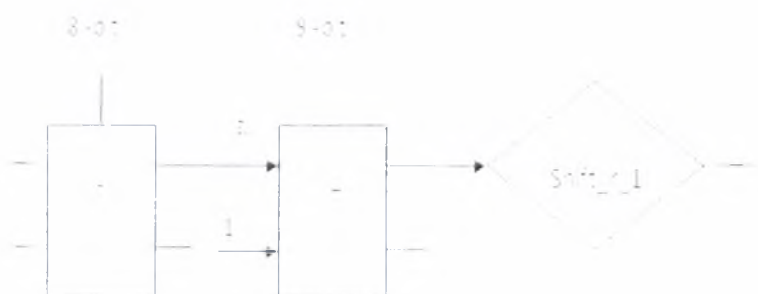
- Για την dc πρόβλεψη όταν όλα τα δείγματα είναι διαθέσιμα (σταθμιακή πρόβλεψη):



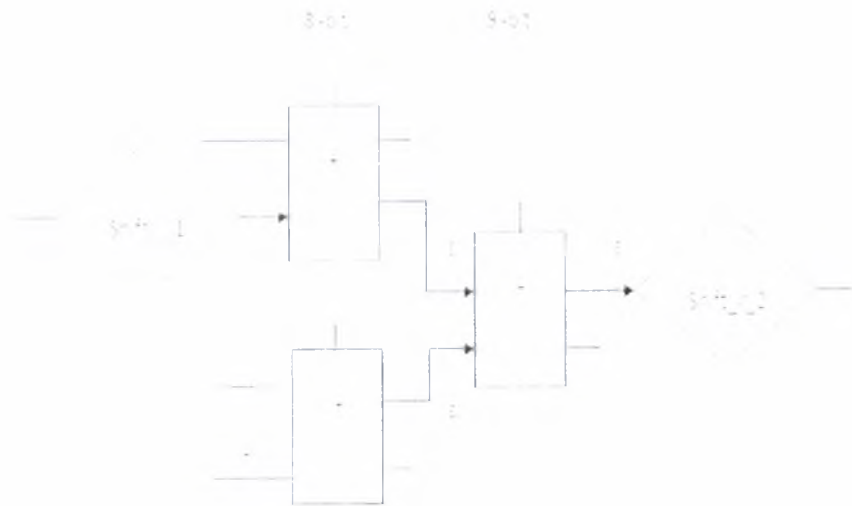
- Για την dc πρόβλεψη όταν είναι διαθέσιμα τα πάνω ή αριστερά δείγματα



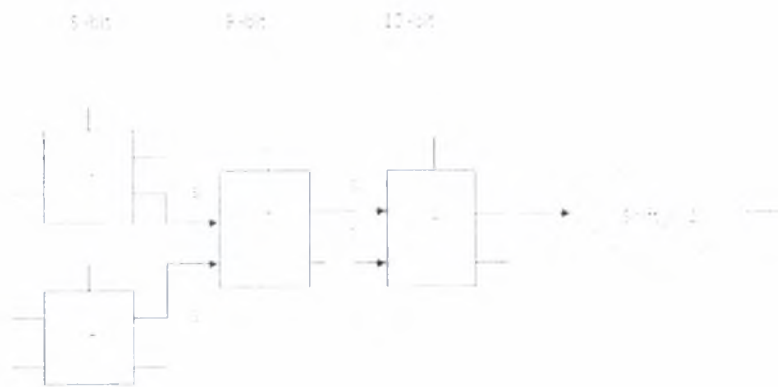
- Για το άθροισμα  $(s(i) + s(j) + 1) \gg 1$



➤ Για το άθροισμα  $(s(i) + 2 * s(j) + s(k) + 2) \gg 2$  :



➤ Για το άθροισμα  $(s(i) + 3 * s(j) + 2) \gg 2$  :



### 3.1. Υλοποίηση μεθόδων πρόβλεψης για τα μπλοκ φαστηνότητας 16X16

Για την υλοποίηση των μεθόδων πρόβλεψης για τα μπλοκ 16X16, η φιλοσοφία της υλοποίησης της  $dc$  συνάρτησης είναι ίδια με αυτή για μπλοκ 4X4 με 32 δείγματα αυτή τη φορά. Όσον αφορά την  $plane$  συνάρτηση, η υλοποίησή της απαιτεί αθροιστές και πολλαπλασιαστές. Αρχικά θα πρέπει να υπολογιστούν τα αθροίσματα  $H$  και  $V$ . Ο Υπολογισμός τους απαιτεί αφαιρέσεις μεταξύ των δειγμάτων, πολλαπλασιασμό του κάθε αποτελέσματος της αφαίρεσης και τέλος πρόσθεση όλων αυτών των αποτελεσμάτων. Έπειτα υπολογίζουμε τις μεταβλητές  $a$ ,  $b$ ,  $c$ .

Εφόσον έχουμε υπολογίσει όλα τα προηγούμενα, δημιουργούμε ένα `component` το οποίο πραγματοποιεί τις προσθέσεις των τεσσάρων εισόδων, και στη συνέχεια αναθέτει στον πίνακα πρόβλεψης την κάθε τιμή.

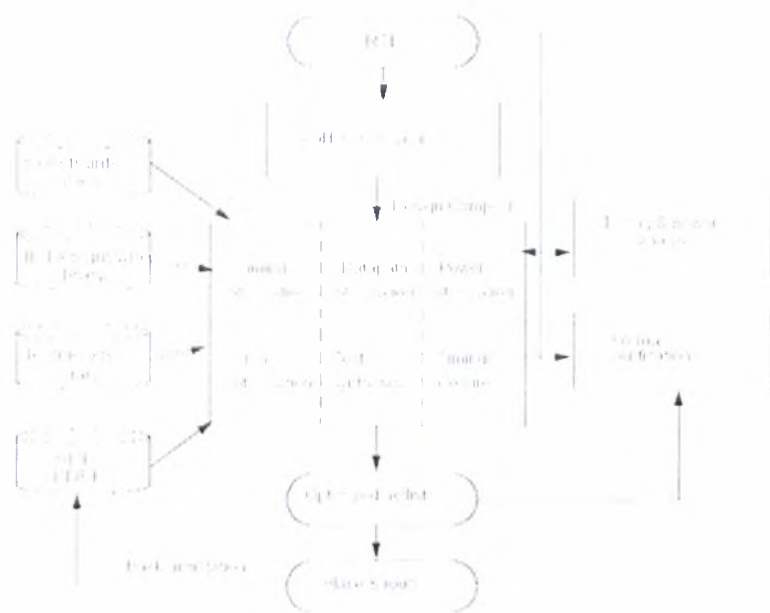
## 1. Εισαγωγή

Για τις ανάγκες της υλοποίησης της διπλωματικής εργασίας, χρησιμοποιήθηκαν εργαλεία της Synopsys τα οποία συντέλεσαν στην πραγματοποίηση πειραματικών μετρήσεων και στην εξαγωγή αποτελεσμάτων. Ένα από αυτά τα εργαλεία είναι ο Design compiler (DeCo).

### 2. Design compiler

Ο Design Compiler (DC) είναι το λογισμικό που καλείται από το σύνολο των προγραμμάτων που προσφέρονται για την σύνθεση ψηφιακών κυκλωμάτων. Ο DC βελτιστοποιεί τις σχεδιάσεις με απώτερο στόχο την παροχή μικρότερων και γρηγορότερων αναπαραστάσεων μιας λογικής συνάρτησης. Περιέχει υποεργαλεία τα οποία συνθέτουν τις HDL σχεδιάσεις σε τεχνολογικά εξαρτημένες σχεδιάσεις σε επίπεδο πυλών. Υποστηρίζει την δυνατότητα βελτιστοποίησης σχεδιάσεων είτε σε ιεραρχική είτε σε flat μορφή και μπορεί να συνθέσει τόσο ακολουθιακά όσο και συνδυαστικά κυκλώματα βελτιώνοντας την ταχύτητα απόκρισής τους, τον χώρο που καταλαμβάνουν και την ισχύ που καταναλώνουν.

Στο [εξώφυλλο](#) της επόμενης σελίδας παρουσιάζεται ο τρόπος με τον οποίο ο DC συμβάλλει στην συνολική ροή σχεδίασης :



Στο σημείο αυτό θα δώσουμε και τον πρώτο «επίσημο» ορισμό της σύνθεσης ενός ψηφιακού κυκλώματος :

*«Σύνθεση ονομάζεται η διαδικασία κατά την οποία μετατρέπουμε μια σχεδίαση, η οποία μας δίνεται σε HDL κώδικα, σε ένα βέλτιστο netlist σε επίπεδο πυλών το οποίο προσδιορίζεται πλήρως από μια τεχνολογική βιβλιοθήκη»*

Τα βήματα που ακολουθούμε κατά την διαδικασία της σύνθεσης ενός κυκλώματος, σε άμεση αντιστοιχία με το παραπάνω σχήμα, είναι τα εξής :

- 1) Δεχόμαστε ως είσοδο, αρχεία τα οποία περιγράφουν το κύκλωμά μας σε επίπεδο καταχωρητών (RTL – Register Transfer Level), δηλαδή αρχεία σε κάποια γλώσσα περιγραφής υλικού (HDL) όπως η Verilog και η VHDL.

- 2) Κατά την σύνθεση ο Design Compiler μεταφράζει την HDL περιγραφή σε συνθετικά στοιχεία της DesignWare βιβλιοθήκης, όπως είναι για παράδειγμα οι αθροιστές και τα γενικά Boolean συνθετικά στοιχεία (τα οποία στο εξής θα αναφέρονται ως generic Boolean components ή πιο απλά GTECH components). Τα GTECH components δεν έχουν πληροφορίες όσον αφορά τον χρονισμό και την δυνατότητα οδήγησης και δεν προσδιορίζονται – αντιστοιχούν σε κάποια τεχνολογική βιβλιοθήκη. Ο DC χρησιμοποιεί τεχνολογικές (technology), συνθετικές (synthetic) και συμβολικές (symbolic) βιβλιοθήκες από το σημείο αυτό και μετέπειτα για την υλοποίηση της σύνθεσης και την παρουσίαση του αποτελέσματος ως γραφική αναπαράσταση.
  
- 3) Αφού γίνει η μετάφραση - μετατροπή του HDL κώδικα σε επίπεδο πυλών, ο DC βελτιστοποιεί την σχεδίαση και αντιστοιχίζει τα στοιχεία που την αποτελούν σε ένα συνδυασμό αποτελούμενο από συγκεκριμένα κελία βιβλιοθηκών, βασιζόμενος στις επιλογές του χρήστη και τους περιορισμούς που αυτός έχει δώσει. Οι περιορισμοί αποτελούν στην ουσία τις σχεδιαστικές απαιτήσεις του χρήστη για τους στόχους που θέλει να πετύχει με βάση την απόδοση της σχεδίασης, δηλαδή αναφέρονται κατά κύριο λόγο στους χωρικούς και χρονικούς περιορισμούς υπό τους οποίους καλείται να επιτελεστεί η σύνθεση.
  
- 4) Αφού ολοκληρωθεί η φάση αυτή, ακολουθώντας την ροή σχεδίασης που παρουσιάστηκε στο παραπάνω σχήμα, ελέγχουμε το αποτέλεσμα της σύνθεσης για να δούμε αν πληρεί τις απαιτήσεις που θέσαμε στην προηγούμενη φάση έτσι ώστε στην περίπτωση που παρουσιαστεί οποιοδήποτε πρόβλημα να το επιλύσουμε σε όσο το δυνατό πιο πρώιμο στάδιο του κύκλου σχεδίασης.



5) Μετά το πέρας του παραπάνω σταδίου, το κύκλωμά μας, όπως έχει πλέον διαμορφωθεί, είναι έτοιμο για την προσαγωγή του στα place & route εργαλεία του πακέτου που παρέχει η Synopsys, τα οποία σε γενικές γραμμές είναι αρμόδια για την τοποθέτηση και την διασύνδεση των επιμέρους κελιών στην σχεδίαση. Ο σχεδιαστής στο σημείο αυτό, έχει την δυνατότητα να επισημειώσει επιπρόσθετα στοιχεία, όπως οι καθυστερήσεις που παρουσιάζονται στις εσωτερικές διασυνδέσεις (interconnection delays) και να βάλει τον Design Compiler να επανασυνθέσει την σχεδίαση προκειμένου η ανάλυση χρόνισμού να είναι πιο ακριβής.

Ο Design Compiler διαβάζει και παράγει αρχεία που προσδιορίζουν σχεδιάσεις σε πολλαπλές μορφές που αποτελούν standards στην ηλεκτρονική σχεδίαση ψηφιακών κυκλωμάτων, συμπεριλαμβανομένων των αρχείων της Synopsys .db και eqn. Επιπρόσθετα ο DC προσφέρει άμεση σύνδεση με άλλα EDA (Electronic Design Automation) εργαλεία. Αυτό μας δίνει την δυνατότητα να μεταφέρουμε με ευκολία από εργαλείο σε εργαλείο τους περιορισμούς και τα αποτελέσματα που έχουμε ανακτήσει μέχρι την προκειμένη στιγμή.

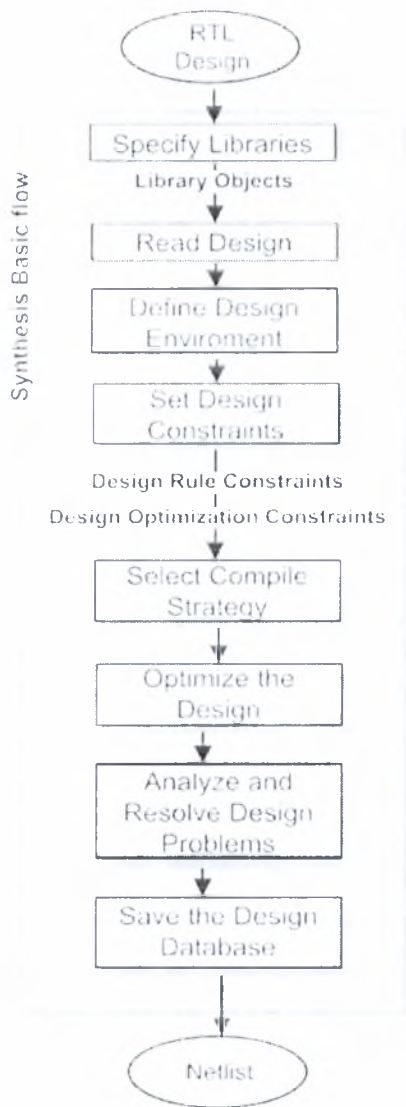
Στην συνέχεια θα αναφερθούμε, πιο αναλυτικά, στα εργαλεία που χρησιμοποιήθηκαν για την σύνθεση των κυκλωμάτων της παρούσας εργασίας (Design Vision) και τις βιβλιοθήκες που παρέχει η Synopsys και συνεικονούν στην επιτυχημένη σύνθεση ψηφιακών κυκλωμάτων.

Το Design Vision είναι μια γραφική διεπαφή (graphical user interface - GUI) για το περιβάλλον σύνθεσης της Synopsys και ένα εργαλείο το οποίο μπορεί να χρησιμοποιηθεί για την ανάλυση της σχεδίασής μας και την «αντιστοίχισή» της στην GTECH τεχνολογική βιβλιοθήκη σε επίπεδο πυλών.

Το γραφικό περιβάλλον του Design Vision περιέχει μενού επιλογών τα οποία μας δίνουν την δυνατότητα να εκτελέσουμε το μεγαλύτερο ποσοστό των

εντολών που μπορούν να εκτελεστούν μέσα από το dc shell. Με άλλα λόγια το εργαλείο αυτό αποτελεί μια μετεξέλιξη του Design Analyzer το οποίο είναι μια ακόμη γραφική διεπαφή που προσφέρεται για την επικοινωνία με τον πυρήνα των εργαλείων της Synopsys. Στο σημείο αυτό πρέπει να αναφέρουμε πως όλες οι δυνατότητες του DC μπορούν να χρησιμοποιηθούν μέσα από την κονσόλα που μας προσφέρει το design vision, και αυτό συμβαίνει διότι τόσο το Design Vision όσο και ο Design Compiler χρησιμοποιούν την ίδια «μηχανή» ανάλυσης στατικού χρονισμού.

Η μεθοδολογία που ακολουθείται είναι όμοια με αυτήν που περιγράψαμε παραπάνω, λόγω του γεγονότος πως το καλούν περιβάλλον είναι αυτό του Design Compiler. Η μεθοδολογία αυτή μπορεί να περιγραφεί από το ακόλουθο σχήμα :



Θα παρουσιάσουμε βήμα – βήμα την πορεία που ακολουθείται βασισμένη στο παρακάτω script που αποτελεί μια γενικευμένη εκδοχή αυτών που χρησιμοποιήθηκαν :

```
set link library {/home/adadalia/UMC013/umce13h210t3 tc 120V 25C.db}
set symbol library {/home/adadalia/UMC013/umce13h210t3 tc 120V 25C.sdb}
set target library {/home/adadalia/UMC013/umce13h210t3 tc 120V 25C.db}
```

```
analyze -format vhdl {/home/adadalia/VASILIS/add/types.vhd}
analyze -format vhdl {/home/adadalia/VASILIS/add/and_gate.vhd}
analyze -format vhdl {/home/adadalia/VASILIS/add/or_gate.vhd}
analyze -format vhdl {/home/adadalia/VASILIS/add/xor_gate.vhd}
analyze -format vhdl {/home/adadalia/VASILIS/add/adder1.vhd}
analyze -format vhdl {/home/adadalia/VASILIS/add/adder8.vhd}
analyze -format vhdl {/home/adadalia/VASILIS/add/adder9.vhd}
analyze -format vhdl {/home/adadalia/VASILIS/add/adder10.vhd}
analyze -format vhdl {/home/adadalia/VASILIS/add/adder11.vhd}
analyze -format vhdl {/home/adadalia/VASILIS/add/intra4x4.vhd}
```

```
elaborate intra_circuit -library WORK
current design intra_circuit
uniquify
set flatten true -design intra_circuit -effort high -minimize single_output -phase
true
set max_delay 1.0 [all outputs]
set max_transition 0.1 [all inputs]
compile -map effort high -ungroup all -exact map
report timing -path full -delay max -nworst 1 -max_paths 1 -significant_digits 3 -
sort_by group >> /home/adadalia/VASILIS/add/time03.txt
report area -nosplit >> /home/adadalia/VASILIS/add/area1.txt
report cell -physical >> /home/adadalia/VASILIS/add/cell1.txt
```

Οι εντολές, με την σειρά που αναγράφονται, αποτελούν ένα τυπικό παράδειγμα ενός script το οποίο οδηγεί το Design Vision στην σύνθεση ενός ψηφιακού κυκλώματος και την αποθήκευση της νέας σχεδίασης στο δίσκο.

Οι «εντολές» “link library” και “target library” προσδιορίζουν την θέση όπου βρίσκονται οι τεχνολογικές βιβλιοθήκες που προσδιορίζουν τα κελιά και άλλες πληροφορίες ανάλογα με τον «παροχέα» της τεχνολογίας, όπως τα ονόματα των κελιών τους σχεδιαστικούς κανόνες και τις συνθήκες λειτουργίας. Η

παράμετρος "symbol library" καθορίζει τα σύμβολα που είναι διαθέσιμα για την δημιουργία των κατάλληλων schematics.

Για να ξεκινήσουμε να δουλεύουμε πάνω στην σχεδίαση, πρέπει αρχικά να διαβάσουμε την σχεδίασή μας από τον δίσκο στην ενεργή μνήμη του εργαλείου. Από το σημείο αυτό και έπειτα θα πραγματοποιηθούν όλες οι αλλαγές πριν η σχεδίασή μας αποθηκευτεί και πάλι στον δίσκο. Υπάρχουν δυο δυνατές επιλογές στην περίπτωση που θέλουμε να διαβάσουμε μια σχεδίαση :

- *Analyze & Elaborate* : Χρησιμοποιούμε αυτές τις εντολές για να διαβάσουμε HDL σχεδιάσεις και να τις μετατρέψουμε σε αρχεία βάσης δεδομένων στο format της Synopsys (.db).
- *Read* : Η εντολή αυτή χρησιμοποιείται όταν θέλουμε να διαβάσουμε σχεδιάσεις που είναι ήδη σε .db μορφή.

Η εντολή "analyze" εξετάζει το HDL αρχείο προκειμένου να δει εάν βρίσκεται σε σωστή συντακτική λογική και αν είναι δυνατό να υποστεί σύνθεση. Επιπρόσθετα μεταφράζει τα αρχεία μας σε ένα ενδιάμεσο format και τοποθετεί το αρχείο αυτό στον φάκελο που έχουμε προσδιορίσει ως φάκελο εργασίας (working directory).

Η εντολή "elaborate" εξετάζει το «ενδιάμεσο» αρχείο και στην συνέχεια δημιουργεί την σχεδίαση σε μορφή .db αρχείου. Κατά την διάρκεια αυτής της διαδικασίας το πρόγραμμα καθορίζει ποια από τα «στοιχεία» της σχεδίασης πρέπει να αντικατασταθούν από συνθετικά «στοιχεία» της βιβλιοθήκης που έχουμε επιλέξει.

Η εντολή "read" αποτελεί ένα συνδυασμό των παραπάνω εντολών, δεν πραγματοποιεί, όμως, όσους ελέγχους πραγματοποιούν οι προκείμενες εντολές.

Ακολουθως καλούμαστε να καθορίσουμε το περιβάλλον της σχεδίασης και τους περιορισμούς/παραμέτρους της σχεδίασης. Ο DC απαιτεί την

μοντελοποίηση του σχεδιαστικού περιβάλλοντος του κυκλώματος ώστε να προχωρήσει η διαδικασία της σύνθεσης. Με το μοντέλο αυτό καθορίζουμε τις εξωτερικές συνθήκες λειτουργίας (manufacturing process, temperature, voltage), loads, drives, fanouts, και wire load models. Ο καθορισμός όλων αυτών των παραμέτρων επηρεάζει άμεσα την σύνθεση του κυκλώματος και τα αποτελέσματα των βελτιστοποιήσεων που εφαρμόζει ο DC.

Τέλος, με χρήση της εντολής "report" και μιας σειράς παράμετρων λαμβάνουμε πληροφορίες για την σχεδίαση, τα κελιά, τον χώρο που καταλαμβάνει το κύκλωμα αλλά και το μέγιστο μονοπάτι που υπάρχει στο κύκλωμα (critical path). Ο υπολογισμός του critical path είναι επιθυμητός διότι εΐθισται ως εμπειρικός κανόνας να τοποθετείται ως περίοδος στα testbenches των κυκλωμάτων η ποσότητα *critical path + 10%critical path*.

### 3. Παιραματικ΄ς μετρησεις και αποτελεσματα

Για το κύκλωμα που υλοποιεί τις μεθόδους πρόβλεψης για μπλοκ 4X4:

Max delay	# of cells	Total cell area	Data arrival time	Max frequency
<b>0.6</b>	<b>2492</b>	<b>27891.64</b>	<b>1.279 ns</b>	<b>780 MHz</b>
0.7	2440	26906.68	1.308 ns	764 MHz
<b>0.9</b>	<b>2430</b>	<b>25824.96</b>	<b>1.298 ns</b>	<b>770 MHz</b>
1.0	2663	28283.87	1.279 ns	780 MHz
1.1	3071	29426.11	1.312 ns	762 MHz
1.2	3165	28548.28	1.289 ns	775 MHz

- [1] <http://www.fishbase.org/summary.php?species=11264>
- [2] <http://www.fishbase.org/summary.php?species=11264>
- [3] <http://www.fishbase.org/summary.php?species=11264>
- [4] <http://www.fishbase.org/summary.php?species=11264>
- [5] <http://www.fishbase.org/summary.php?species=11264>
- [6] <http://www.fishbase.org/summary.php?species=11264>
- [7] <http://www.fishbase.org/summary.php?species=11264>
- [8] <http://www.fishbase.org/summary.php?species=11264>
- [9] <http://www.fishbase.org/summary.php?species=11264>
- [10] <http://www.fishbase.org/summary.php?species=11264>
- [11] <http://www.fishbase.org/summary.php?species=11264>
- [12] <http://www.fishbase.org/summary.php?species=11264>
- [13] <http://www.fishbase.org/summary.php?species=11264>
- [14] <http://www.fishbase.org/summary.php?species=11264>
- [15] <http://www.fishbase.org/summary.php?species=11264>

- [16] [https://doi.org/10.1016/j.jmb.2017.05.017](#)
- [17] [https://doi.org/10.1016/j.jmb.2017.05.017](#)
- [18] [https://doi.org/10.1016/j.jmb.2017.05.017](#)
- [19] [https://doi.org/10.1016/j.jmb.2017.05.017](#)
- [20] [https://doi.org/10.1016/j.jmb.2017.05.017](#)
- [21] [https://doi.org/10.1016/j.jmb.2017.05.017](#)
- [22] [https://doi.org/10.1016/j.jmb.2017.05.017](#)







ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΙΑΣ



004000085987

