



**Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών**
Τμήμα Φυσικής
Τομέας Ηλεκτρονικής, Υπολογιστών,
Τηλεπικοινωνιών και Αυτοματισμού

Διπλωματική Εργασία

Μ.Δ.Ε. ΣΤΟΝ ΗΛΕΚΤΡΟΝΙΚΟ ΑΥΤΟΜΑΤΙΣΜΟ

“Παράλληλη επεξεργασία μετασχηματισμού Fourier σε αρχιτεκτονική με συγχωνευμένες μνήμες”

Συγγραφή: Βασίλειος Κιτσάκης ΑΜ: 2014509
Επιβλέπων: Διονύσιος Ρεΐσης, Αναπλ. Καθηγητής ΕΚΠΑ

Τριμελής επιτροπή: Διονύσιος Ρεΐσης, Αναπλ. Καθηγητής ΕΚΠΑ
Ηλίας Μανωλάκος, Καθηγητής ΕΚΠΑ
Μανώλης Τσίλης, Επίκ. Καθηγητής ΕΚΠΑ

Αθήνα, 2017

ΠΕΡΙΛΗΨΗ

Ο γρήγορος μετασχηματισμός Fourier (Fast Fourier Transform – FFT) είναι ένας αλγόριθμος για τον υπολογισμό του διακριτού μετασχηματισμού Fourier (Discrete Fourier Transform – DFT). Λόγο της διαδεδομένης χρήσης του σε διάφορες εφαρμογές, απαιτείται η υλοποίηση του σε υλισμικό. Κατά συνέπεια σε αρκετές εφαρμογές είναι αναγκαία η υλοποίηση τους με το μικρότερο δυνατόν κόστος. Στην παρούσα εργασία παρουσιάζεται μια τεχνική υπολογισμού του FFT με συγχωνευμένη μνήμη (δύο στοιχεία ανά θέση μνήμης), η υλοποίηση της, καθώς και η σύγκρισή της με άλλες τεχνικές που προσπαθούν να επιτύχουν το ελάχιστο κόστος υλοποίησης.

SUMMARY

The Fast Fourier Transform (FFT) is an algorithm for computing the Discrete Fourier Transform (DFT). FFT is used in various applications, thus hardware implementations are vastly used. Many applications though need low cost implementations. This thesis introduces a technique for computing FFT with merged memories (two FFT points per memory shell), the implementation of this technique and it's comparison with other techniques that try to achieve the same goal, low cost implementation.

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ.....	3
2. ΜΙΓΑΔΙΚΟΙ ΑΡΙΘΜΟΙ ΣΕ ΥΛΙΣΜΙΚΟ.....	5
3. ΑΡΙΘΜΗΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΕ ΥΛΙΣΜΙΚΟ.....	9
4. ΜΑΘΗΜΑΤΙΚΟΙ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ.....	12
4.1. ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER.....	12
4.2. ΔΙΑΚΡΙΤΟΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER.....	12
4.3. ΓΡΗΓΟΡΟΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER.....	13
4.3.1.ΔΙΑΙΡΕΣΗ ΣΤΟ ΧΡΟΝΟ.....	13
4.3.2.ΔΙΑΙΡΕΣΗ ΣΤΗ ΣΥΧΝΟΤΗΤΑ.....	17
5. ΘΕΩΡΗΤΙΚΗ ΑΝΑΛΥΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ.....	20
6. ΥΛΟΠΟΙΗΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ.....	22
7. ΠΑΡΑΔΕΙΓΜΑ ΛΕΙΤΟΥΡΓΕΙΑΣ.....	30
8. ΑΠΟΤΕΛΕΣΜΑΤΑ.....	35
9. ΑΝΑΦΟΡΕΣ.....	37

1. ΕΙΣΑΓΩΓΗ

Πληθώρα εφαρμογών στην επεξεργασία σήματος και στις τηλεπικοινωνίες καταστούν αναγκαία την εφαρμογή υπολογισμού του γρήγορου μετασχηματισμού Fourier (Fast Fourier Transform – FFT) με χαμηλό κόστος υλοποίησης. Για να διατηρηθεί το χαμηλό κόστος είναι συνήθεις τεχνική να χρησιμοποιείται ένας επεξεργαστής υπολογισμού της πεταλούδας, βάσης δύο και μια μνήμη μεγέθους ίσου με τον αριθμό των σημείων του μετασχηματισμού (N). Στην μνήμη αυτή, με την χρήση της τεχνικής “in-place”, αποθηκεύονται οι είσοδοι, οι ενδιάμεσοι υπολογισμοί και οι έξοδοι. Με στόχο την βελτιστοποίηση της επεξεργασίας υπάρχει μια επικέντρωση σε τεχνικές διευθυνσιοδότησης οι οποίες επιτρέπουν την φόρτωση και την αποθήκευση παράλληλα των δύο εισόδων και δύο εξόδων της κάθε πεταλούδας (βάσης δύο) [3],[4]. Πολλές τεχνικές το επιτυγχάνουν αυτό χωρίζοντας την μνήμη σε δύο μνήμες (τράπεζες) μεγέθους $N/2$. Έτσι επιτρέπουν την φόρτωση ή την αποθήκευση ενός δεδομένου σε κάθε τράπεζα στον ίδιο κύκλο. Τέτοιες τεχνικές στοχεύουν στην ελαχιστοποίηση του κόστους της παραγωγής των διευθύνσεων και γενικά του data-path. Η τεχνική [4] παρουσιάζει μια λύση για βάση δύο με την χρήση καταχωρητών εξόδου έτσι ώστε να παρακαμφθεί η “διαμάχη” κατά την αποθήκευση των αποτελεσμάτων της πεταλούδας. Η τεχνική [8] δείχνει το σύνολο των αναδιατάξεων (permutations) που παρέχουν μια λύση στην παράλληλη πρόσβαση σε N σημεία FFT υπολογισμών σε βάση b με b τράπεζες και απαιτούνται $\log_b^2 N$ bit look-up-tables (LUTs) για να υλοποιηθούν οι αναδιατάξεις. Η τεχνική [5] προτείνει μια ευρετική προσέγγιση και η [9] μια παράλληλη τεχνική διευθυνσιοδότησης αξιοποιώντας τις ιδιότητες του Gray code. Οι τεχνικές βάσεις τέσσερα [6] και [14] αποφεύγουν την χρήση αναδιατάξεων. Όπου στην [6] χρησιμοποιούνται 4 τράπεζες και 4^2 καταχωρητές στην πεταλούδα για να αποθηκεύονται τέσσερις τετράδες για να εκτελεστούν τέσσερις πεταλούδες. Στην τεχνική [13] χρησιμοποιούνται αναδιατάξεις σε πολλαπλά δεδομένα που παραχωρήθηκαν στον επεξεργαστή για υπολογισμούς μικτής βάσης.

Η παρούσα εργασία προτείνει μια τεχνική διευθυνσιοδότησης για παράλληλη πρόσβαση των δεδομένων με βάση δύο. Αρχικά γίνεται μια εισαγωγή στις μαθηματικές έννοιες, οι οποίες είναι αναγκαίες για την κατανόηση της τεχνικής και του μετασχηματισμού γενικότερα, και πως αυτές υλοποιούνται-αναπαριστάνονται σε εφαρμογές υλισμικού και σε προγράμματα

υλοποιημένα με γλώσσες χαμηλού επιπέδου. Ύστερα γίνεται η ανάλυση της τεχνικής και η υλοποίηση της. Η καινοτομία της προτεινόμενης τεχνικής είναι ότι τα δεδομένα είναι αποθηκευμένα σε ζευγάρια σε κάθε θέση μνήμης σε μια μνήμη $N/2$ θέσεων (συγχωνευμένη μνήμη). Η λογική είναι να καλούνται δύο ζεύγη και να εκτελείται μια ανακατάταξη εισόδου (της πεταλούδας) σε αυτά τα τέσσερα δεδομένα. Τα τέσσερα αποτελέσματα υπόκεινται μια ανακατάταξη εξόδου (ελέγχετε αν τα δεδομένα είναι είσοδοι του FFT ή ενδιάμεσοι υπολογισμοί) πριν αποθηκευτούν πάλι σαν ζεύγη.

2. ΜΙΓΑΔΙΚΟΙ ΑΡΙΘΜΟΙ ΣΕ ΥΛΙΣΜΙΚΟ

Μιγαδικός αριθμός ορίζεται ένας αριθμός που έχει την μορφή $\alpha + \beta i$ με $i^2 = -1$. Το α ορίζεται ως το πραγματικό μέρος ενώ το β ως το φανταστικό. Κατά την άθροιση ισχύει η προσεταιριστική ιδιότητα

$$(\alpha + \beta i) + (\gamma + \delta i) = (\alpha + \gamma) + (\beta + \delta)i$$

για να παραχθεί ο καινούριος μιγαδικός αριθμός χρειάζονται δύο αθροίσεις, μια για το πραγματικό κομμάτι ($\alpha + \beta$) και μια για το φανταστικό κομμάτι ($\gamma + \delta$). Έτσι για να αναπτυχθεί σε κύκλωμα ή σε πρόγραμμα ένας μιγαδικός αθροιστής χρειάζονται δύο αθροιστές, ένας για κάθε κομμάτι όπως αναφέρθηκε.

Αντίστοιχα για την αφαίρεση

$$(\alpha + \beta i) - (\gamma + \delta i) = (\alpha - \gamma) + (\beta - \delta)i$$

για να παραχθεί ο καινούριος μιγαδικός αριθμός χρειάζονται δύο αφαιρέσεις, μια για το πραγματικό κομμάτι ($\alpha - \beta$) και μια για το φανταστικό κομμάτι ($\gamma - \delta$).

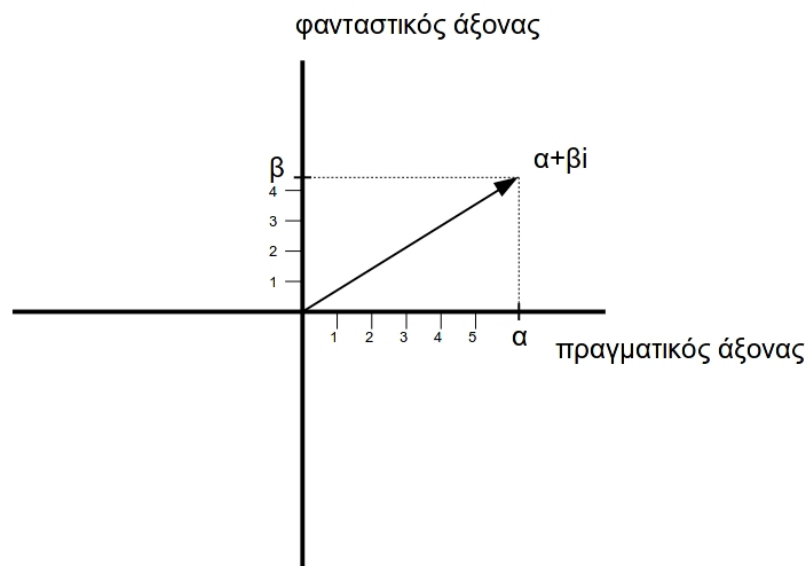
Κατά τον πολλαπλασιασμό ισχύει η επιμεριστική ιδιότητα.

$$\begin{aligned}(\alpha + \beta i) * (\gamma + \delta i) &= \alpha\gamma + \alpha(\delta i) + (\beta i)\gamma + (\beta i)(\delta i) \\&= \alpha\gamma + (\alpha\delta + \beta\gamma)i + (\beta\delta)(i^2) \\&= \alpha\gamma + (\alpha\delta + \beta\gamma)i - \beta\delta \\&= (\alpha\gamma - \beta\delta) + (\alpha\delta + \beta\gamma)i \\&= (\alpha*\gamma - \beta*\delta) + (\alpha*\delta + \beta*\gamma)i\end{aligned}$$

για να παραχθεί ο μιγαδικός αριθμός χρειάζονται για το πραγματικό μέρος δύο πολλαπλασιασμοί και μια αφαίρεση ($\alpha*\gamma - \beta*\delta$) και για το φανταστικό μέρος δύο πολλαπλασιασμοί και μια πρόσθεση

($\alpha*\delta + \beta*\gamma$). Συνολικά τέσσερις πολλαπλασιασμοί, μια πρόσθεση και μια αφαίρεση συνθέτουν τον μιγαδικό πολλαπλασιασμό. Έτσι για να αναπτυχθεί σε κύκλωμα ή σε πρόγραμμα ένας μιγαδικός πολλαπλασιαστής χρειάζονται τέσσερις πολλαπλασιαστές ένας αθροιστής και ένας αφαιρέτης.

Οι μιγαδικοί αριθμοί μπορούν να αναπαρασταθούν σαν διανύσματα στο μιγαδικό πεδίο. Το μιγαδικό πεδίο έχει ως βάση τα διανύσματα $(1,0)$ και $(0,i)$, τα οποία ορίζουν τον πραγματικό άξονα και τον φανταστικό άξονα.



Έστω ο μιγαδικός αριθμός $z = \alpha + \beta i$ σαν μέτρο του ορίζετε

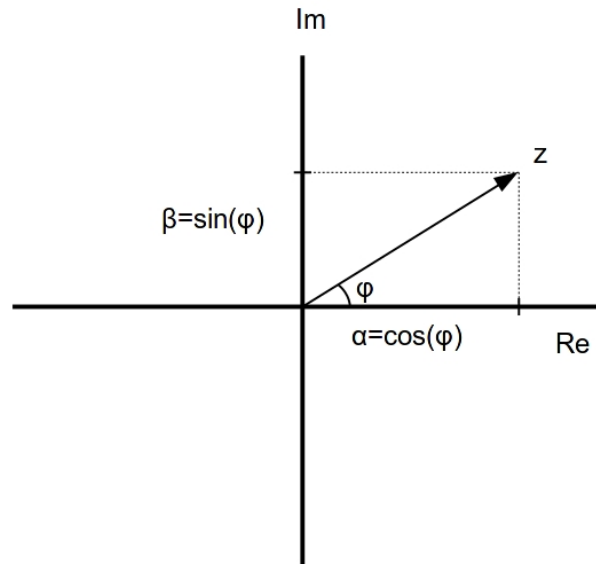
$$|z|^2 = |\alpha + \beta i|^2 = \alpha^2 + \beta^2$$

και σαν γωνία (φάση) το

$$\varphi(z) = \varphi(\alpha + \beta i) = \tan^{-1}(\beta/\alpha)$$

Σύμφωνα με τον τύπο του Euler, ένας μιγαδικός αριθμός μπορεί να γραφεί ως

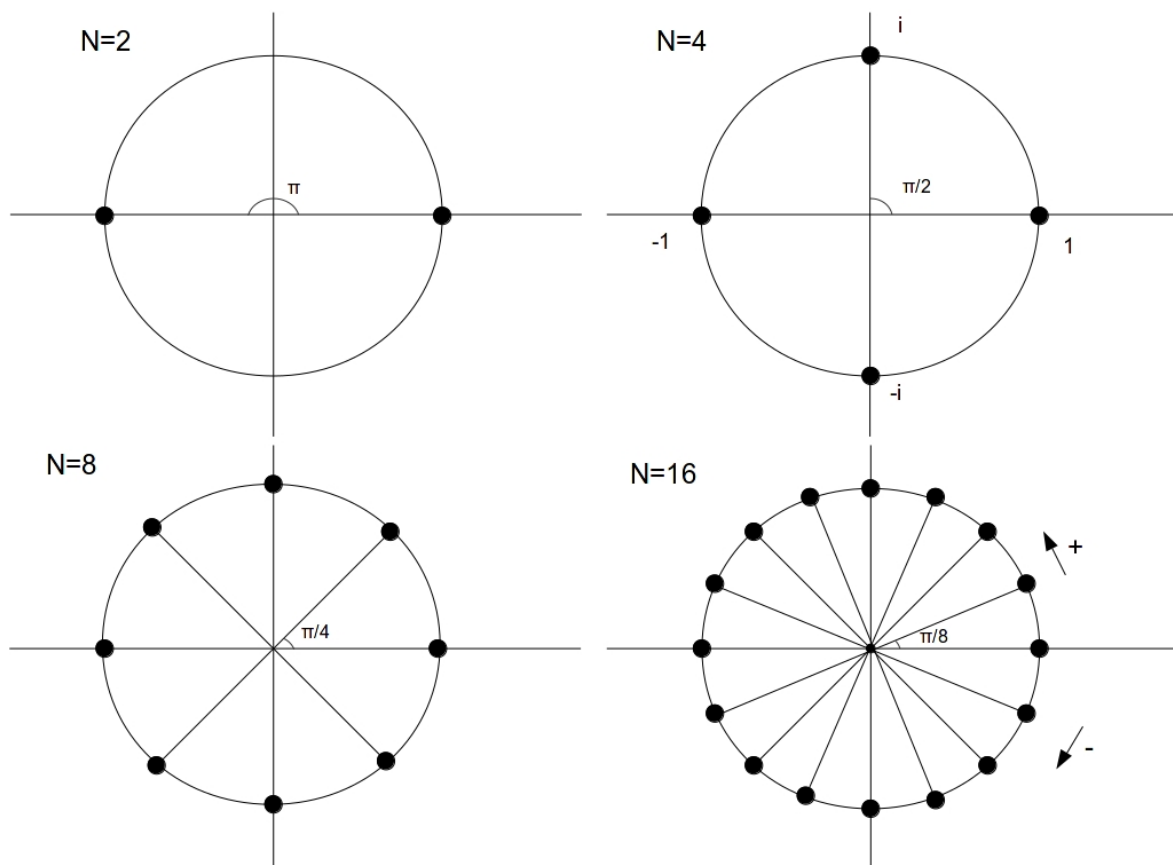
$$z = |z| e^{i\varphi(z)} = |z|\cos(\varphi) + |z|\sin(\varphi)i = \alpha + \beta i, \text{ όπου } \varphi \text{ είναι η γωνία που σχηματίζει το διάνυσμα με τον πραγματικό άξονα.}$$



Δηλαδή για να αναπαρασταθεί ένας μιγαδικός αριθμός σε ένα κύκλωμα (ή σε πρόγραμμα/αλγόριθμο) χρειάζονται δύο μεταβλητές/πληροφορίες είτε το α και β , είτε το $|z|$ και φ . Αν υποθέσουμε ότι το μέτρο του z είναι 1, τότε όλα τα σημεία που επιβιώνουν συνθέτουν τον μοναδιαίο κύκλο.

Η λύση του $z^N=1$ δίνει τις N ρίζες οι οποίες είναι κατανεμημένες πάνω στον μοναδιαίο κύκλο και απέχουν κατά $2\pi/N$ η κάθε μια με την επόμενη της. Έστω $n=0,1,2,\dots,N-1$, τότε το όρισμα του $e^{\pm 2\pi i n/N}$ ανάλογα με το πρόσημό του και την τιμή του n μας δίνει την θέση της αντίστοιχης ρίζας. Αν το πρόσημο είναι αρνητικό τότε οι ρίζες ακολουθούν τη φορά του ρολογιού, εάν είναι θετικό την αντίθετη φορά.

Οι λύσεις αυτές λέγονται και νιοστές ρίζες της μονάδας και έχουν την μορφή $1, \kappa, \kappa^2, \kappa^3, \kappa^4, \dots, \kappa^{N-1}$ όπου $\kappa = e^{2\pi i/N} = \cos(2\pi/N) + i\sin(2\pi/N)$.



3. ΑΡΙΘΜΗΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΕ ΥΛΙΣΜΙΚΟ

Στον κόσμο των κυκλωμάτων μια δυαδική λέξη μπορεί να αναπαριστά έναν ακέραιο αριθμό. Με δύο μπορεί να αναπαρασταθεί ένας μιγαδικός, όπως αναφέρθηκε παραπάνω. Οι αριθμοί όμως που συνθέτουν τους μιγαδικούς αριθμούς δεν είναι ακέραιοι είναι πραγματικοί. Ένας τρόπος να αναπαριστά μια δυαδική λέξη έναν πραγματικό αριθμό είναι η fixed-point αναπαράσταση. Στην αναπαράσταση αυτή υπάρχει μια σταθερή (fixed) θέση που αναπαριστά την υποδιαστολή. Η υποδιαστολή χωρίζει την λέξη σε δύο κομμάτια. Το αριστερό κομμάτι αναπαριστά τις μονάδες και το δεξί τα κλάσματα. Στην πραγματικότητα ένας fixed-point αριθμός είναι ένας ακέραιος, αρκεί να αναχθεί σε έναν ακέραιο με κάποιον παράγοντα αναγωγής. Έστω ο αριθμός 2.5 από το σύνολο των δεκαδικών αριθμών, και έστω ο παράγοντας αναγωγής 10 από το ίδιο σύνολο. Πολλαπλασιάζοντας το 2.5 με το 10 το αποτέλεσμα είναι 25. Από τον κλασματικό (έστω fixed-point) 2.5 κατέληξε στον ακέραιο 25. Έτσι σύμφωνα με τον παράγοντα αναγωγής 10 το 25 αναπαριστά το 2.5. Το 25 πλέον μπορεί να χρησιμοποιηθεί σαν ακέραιος (ακέραια αριθμητική). Έστω οι αριθμοί 1.3 και 1.4 από το σύνολο των δεκαδικών αριθμών. Η πρόσθεση τους δίνει $1.3 + 1.4 = 2.7$. Με αναγωγή με παράγοντα το 10 μετασχηματίζεται σε $13 + 14 = 27$, όπου η πρόσθεση είναι πρόσθεση ακεραίων, το αποτέλεσμα πρέπει να αναχθεί βέβαια σε 2.7 για να βρεθεί στην σωστή αναπαράσταση.

Τα βήματα είναι τα εξείς

- 1) $1.3 \Rightarrow 13$
- 2) $1.4 \Rightarrow 14$
- 3) $13 + 14 = 27$
- 4) $27 \Rightarrow 2.7$

Όπως φαίνεται από τα προηγούμενα ο παράγοντας αναγωγής 10 είναι πολύ ικανοποιητικός μιας και η αναπαράσταση των δεκαδικών αριθμών γίνεται με βάση το 10. Έτσι στους δεκαδικούς αριθμούς διαλέγοντας σαν παράγοντα αναγωγής την κατάλληλη δύναμη του 10, μπορεί να μετασχηματιστεί ο αριθμός σε ακέραιο. Με το παραπάνω είναι προφανές ότι στους δυαδικούς αριθμούς οι καταλληλότεροι παράγοντες αναγωγής είναι οι δυνάμεις του 2.

Έστω οι fixed-point αριθμοί 001.10 και 001.01 είναι οι 1.5 και 1.25 αντίστοιχα. Αριστερά της υποδιαστολής το κάθε bit είναι η αντίστοιχη δύναμη του 2 ενώ δεξιά η αντίστοιχη αρνητική δύναμη του 2. Άρα $001.10 = 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} = 1.5$ και

$001.01 = 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 1.25$. Το άθροισμά τους είναι 2.75. Τα κλασματικά ψηφία είναι 2 έτσι σαν παράγοντα αναγωγής μπορώ να διαλέξω το 2^2 . Έτσι αντί για 001.10 (1.5) μπορώ να έχω 00110 ($1.5 \cdot 4 = 6$) και αντί για 001.01 (1.25) μπορώ να έχω 00101 ($1.25 \cdot 4 = 5$). Το αποτέλεσμα της άθροισης είναι 01011 που είναι το 11 σε δεκαδική αναπαράσταση. Εάν πάμε από το 01011 στο 010.11 είναι σαν να διαιρούμε το 11 με το 4, δηλαδή κάνουμε την αναγωγή στην αρχική μορφή. Όμως $11/4 = 2.75$ που συμφωνεί με την αναπαράσταση $010.11 = 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} +$

$1 \cdot 2^{-2} = 2.75$. Η αναπαράσταση του 010.11 συμβολίζεται ως Q2.2 που σημαίνει το πρώτο bit είναι το πρόσημο, τα επόμενα δύο το ακέραιο κομμάτι και τέλος το τελευταία δύο το κλασματικό κομμάτι.

Είναι προφανές ότι η πρόσθεση είναι ακριβώς ίδια και στις δύο περιπτώσεις. Δεν ισχύει όμως το ίδιο στην περίπτωση του πολλαπλασιασμού. Έστω οι δεκαδικοί αριθμοί 1.5 και 2, ισχύει ότι $1.5 \cdot 2 = 3$, με παράγοντα το 4 έχουμε $6 \cdot 8 = 48$. Είναι πασιφανές ότι ο παράγοντας για να επιστρέψει πίσω στους κλασματικούς δεν είναι πλέον το 4 ($48/4 = 12 \neq 3$) . Αυτό συμβαίνει επειδή ο παράγοντας πολλαπλασιάζεται με τον εαυτό του ($4 \cdot 4 = 16$), έτσι ο παράγοντας “επιστροφής” είναι ο 16 ($48 / 16 = 3$) . Στο δυαδικό σύνολο η παραπάνω διαδικασία έχει ως εξής:

Το 1.5 αναπαριστάται ως 001.10 και το 2 ως 010.00, μπορούν να αναχθούν σε 6 ή 00110 και 8 ή 01000 μέσω του 2^2 (στο τετράγωνο επειδή τα κλασματικά ψηφία είναι δύο) . Ο πολλαπλασιασμός γίνεται βάση της επιμεριστικής ιδιότητας που ισχύει και στους δεκαδικούς, αλλά σε κάθε στοιχείο γίνεται επέκταση πρόσημου (αριστερότερο ψηφίο).

						0	0	1	1	0
					x	0	1	0	0	0
				0	0	0	0	0	0	0
			0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0
	0	0	0	0	1	1	0			
+	0	0	0	0	0	0				
	0	0	0	0	1	1	0	0	0	0

Η κάθε λέξη είχε 2 κλασματικά bit, οπότε το αποτέλεσμα έχει 4 κλασματικά bit. Αντίστοιχα το ακέραιο κομμάτι είχε 3 bit, άρα το αποτέλεσμα έχει 6 bit. Συνεπώς το αποτέλεσμα μπορεί να διαβαστεί ως $000110000 = 48$ ή $00011.0000 = 3$ μετά την αναγωγή. Μέσω στρογγυλοποίησης μπορεί το αποτέλεσμα να μετασχηματιστεί από την μορφή Q5.4 στην μορφή Q2.2, που είναι ίδια με αυτήν που είχαν οι αριθμοί που πολλαπλασιάστηκαν. Μπορεί να παρατηρηθεί ότι ο αριθμός 011.00 είναι αρκετός για να αναπαρασταθεί το 3, είναι μικρότερο το μήκος/κόστος της λέξης θυσιάζοντας όμως την ακρίβεια της αναπαράστασης. Στο κύκλωμα που αναπτύχθηκε, όπως θα φανεί παρακάτω, χρησιμοποιήθηκε η αναπαράσταση Q23 (ένα bit το πρόσημο 23 bit τα κλασματικά ψηφία).

4. ΜΑΘΗΜΑΤΙΚΟΙ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΙ

Οι συναρτήσεις μετασχηματισμού είναι ένα μαθηματικό εργαλείο με το οποίο μπορεί να μεταφερθεί μία συνάρτηση από ένα πεδίο σε ένα άλλο. Μαθηματικά παρουσιάζουν ομοιότητες με τους τελεστές, επειδή δρουν πάνω σε μια συνάρτηση και παράγουν μια άλλη, και έχουν όμοιες ιδιότητες με τους τελεστές. Οι μετασχηματισμοί αυτοί είναι αντιστρέψιμοι, υπάρχουν οι ορθοί μετασχηματισμοί και οι ανάστροφοι μετασχηματισμοί. Ένας από αυτούς είναι ο μετασχηματισμός Fourier που χρησιμοποιείται για την μεταφορά μιας συνάρτησης από το πεδίο του χρόνου στο πεδίο της συχνότητας, είτε για το ανάστροφο από το πεδίο της συχνότητας στο πεδίο του χρόνου (ανάστροφος μετασχηματισμός Fourier).

4.1. ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER (Fourier Transform - FT)

Κάθε περιοδικό σήμα μπορεί να αναλυθεί σε άθροισμα μετατοπισμένων στην φάση ημίτονων και συνημίτονων διαφορετικών συχνοτήτων που είναι ακέραιες πολλαπλάσιες μιας θεμελιώδους. Μπορούν να αναλυθούν σε σειρές Fourier. Ο μετασχηματισμός Fourier είναι η επέκταση των σειρών Fourier όταν η περίοδος της συνάρτησης/σήματος τείνει στο άπειρο.

4.2. ΔΙΑΚΡΙΤΟΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER (Discrete Fourier Transform - DFT)

Ο διακριτός μετασχηματισμός Fourier (DFT) μετατρέπει μια πεπερασμένη σειρά από ίσο-κατανεμημένα στοιχεία/δείγματα μίας συνάρτησης/σήματος σε μία ίσου μήκους ίσο-κατανεμημένη ακολουθία του διακριτού χρόνου μετασχηματισμού Fourier (DTFT), η οποία είναι μια ακολουθία συντελεστών από ένα πεπερασμένο συνδυασμό ημιτονοειδών μιγαδικών αριθμών καθορισμένων από τις συχνότητές τους.

4.3. ΓΡΗΓΟΡΟΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ FOURIER (Fast Fourier Transform - FFT)

Ο γρήγορος μετασχηματισμός Fourier (FFT) είναι ένας αλγόριθμος υπολογισμού του DFT ο οποίος μειώνει την πολυπλοκότητα των υπολογισμών από $O(N^2)$ σε $O(N \log_2 N)$.

4.3.1. ΔΙΑΙΡΕΣΗ ΣΤΟ ΧΡΟΝΟ Decimation In Time (DIT)

Έστω

$$W_N = e^{\frac{-i2\pi}{N}}$$

Τότε ο DFT μπορεί να γραφεί ως

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}$$

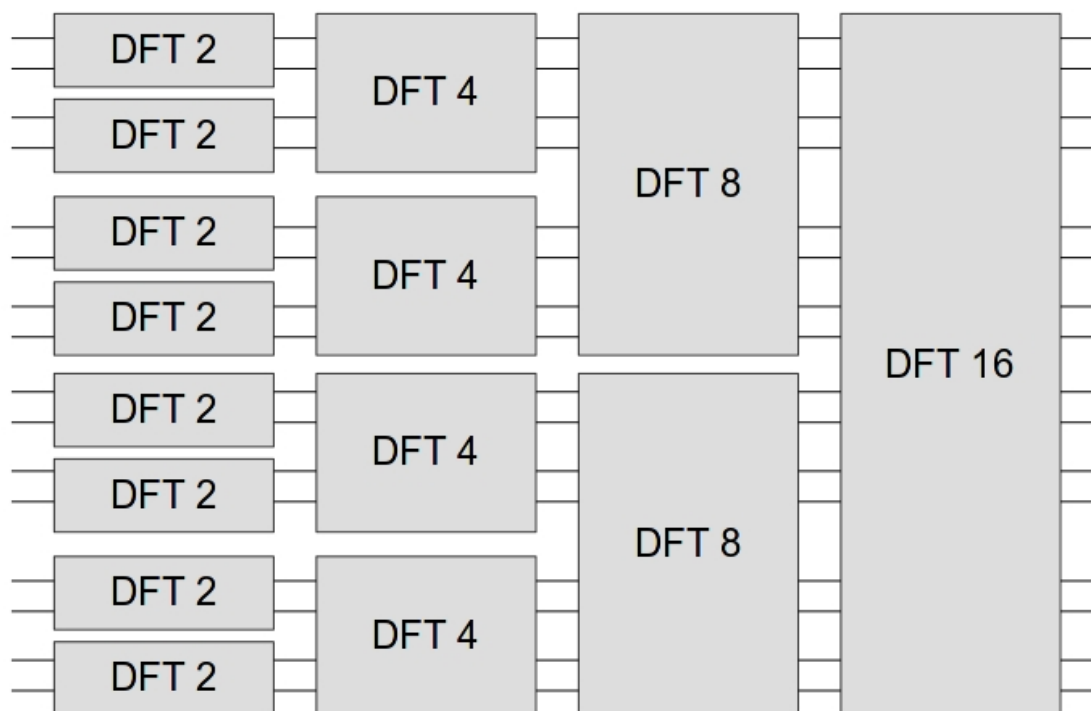
$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}$$

$$X(k) = \sum_{l=0}^{N/2-1} x(2l) W_N^{2lk} + \sum_{l=0}^{N/2-1} x(2l+1) W_N^{(2l+1)k}$$

$$X(k) = \sum_{l=0}^{N/2-1} x(2l) [W_N^2]^{lk} + W_N^k \sum_{l=0}^{N/2-1} x(2l+1) [W_N^2]^{lk}$$

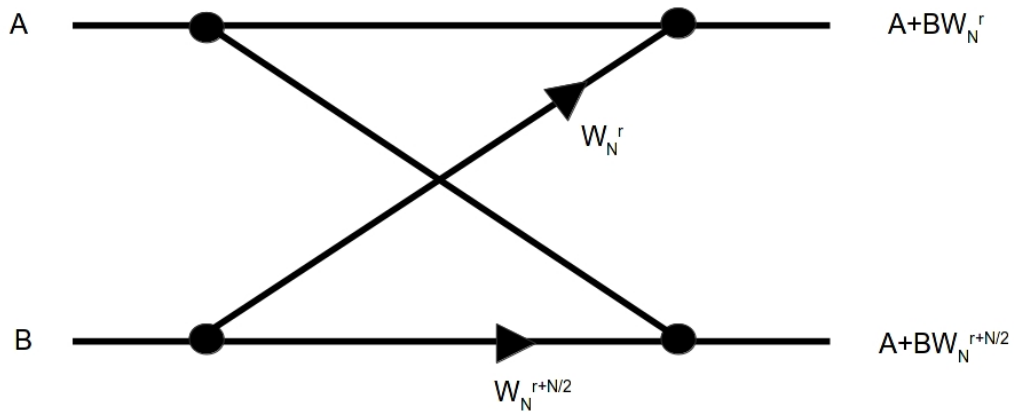
$$X(k) = \sum_{l=0}^{N/2-1} x(2l) [W_{N/2}]^{lk} + W_N^k \sum_{l=0}^{N/2-1} x(2l+1) [W_{N/2}]^{lk}$$

μπορούμε να χωρίσουμε τον χρόνο n σε άρτιους ($2l$) και σε περιττούς ($2l+1$) έτσι το άθροισμα θα χωριστεί σε δύο μέρη.

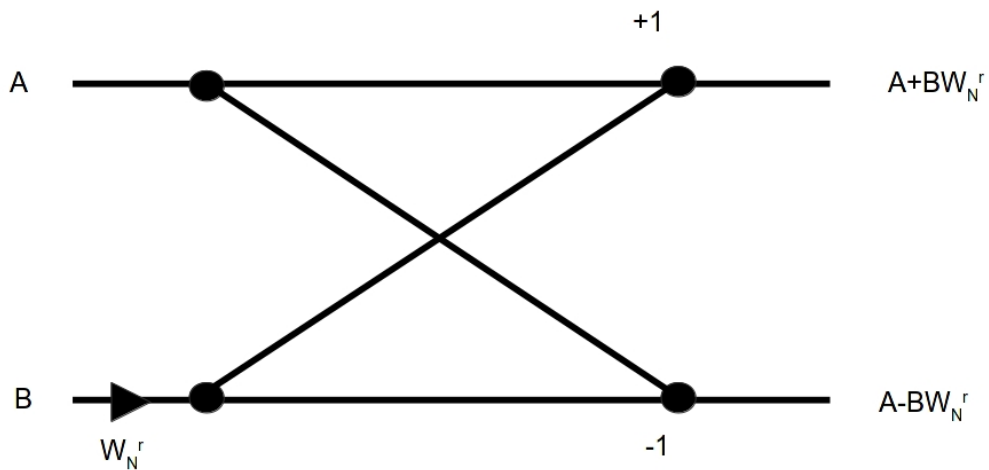


Έτσι κατέληξε σε δύο αθροίσματα όπου το κάθε άθροισμα είναι ένας DFT $N/2$ σημείων, όπου ο δεύτερος είναι πολλαπλασιασμένος με τις σταθερές W_N . Σύμφωνα με αυτό κάθε DFT άρτιου N μπορεί να αναλυθεί σε δύο DFT $N/2$ σημείων. Με τη σειρά τους ο κάθε $N/2$ DFT αν είναι άρτιος μπορεί να αναλυθεί σε δύο DFT $N/4$ σημείων. Επαγωγικά μπορούμε να καταλήξουμε σε DFT 2 σημείων. Αποτέλεσμα είναι ένα δέντρο από DFT ύψους $\log_2 N$, με κάθε επίπεδο να αποτελείται από DFT.

Όπως αναφέρθηκε πριν κάθε σημείο είναι το άθροισμα του αντίστοιχου σημείου μιας ακολουθίας με το αντίστοιχο μιας άλλης επί την ανάλογη σταθερά. Η αναπαράσταση αυτή γίνεται με τις λεγόμενες πεταλούδες. Οι πεταλούδες αυτές βρίσκονται σε όλα τα επίπεδα, δέχονται δύο εισόδους (σημεία) και παράγουν δύο εξόδους. Η μια έξοδος είναι η πρώτη είσοδος συν την δεύτερη πολλαπλασιασμένη με μία σταθερά και η άλλη έξοδος είναι η πρώτη συν την δεύτερη πολλαπλασιασμένη με μια άλλη σταθερά.

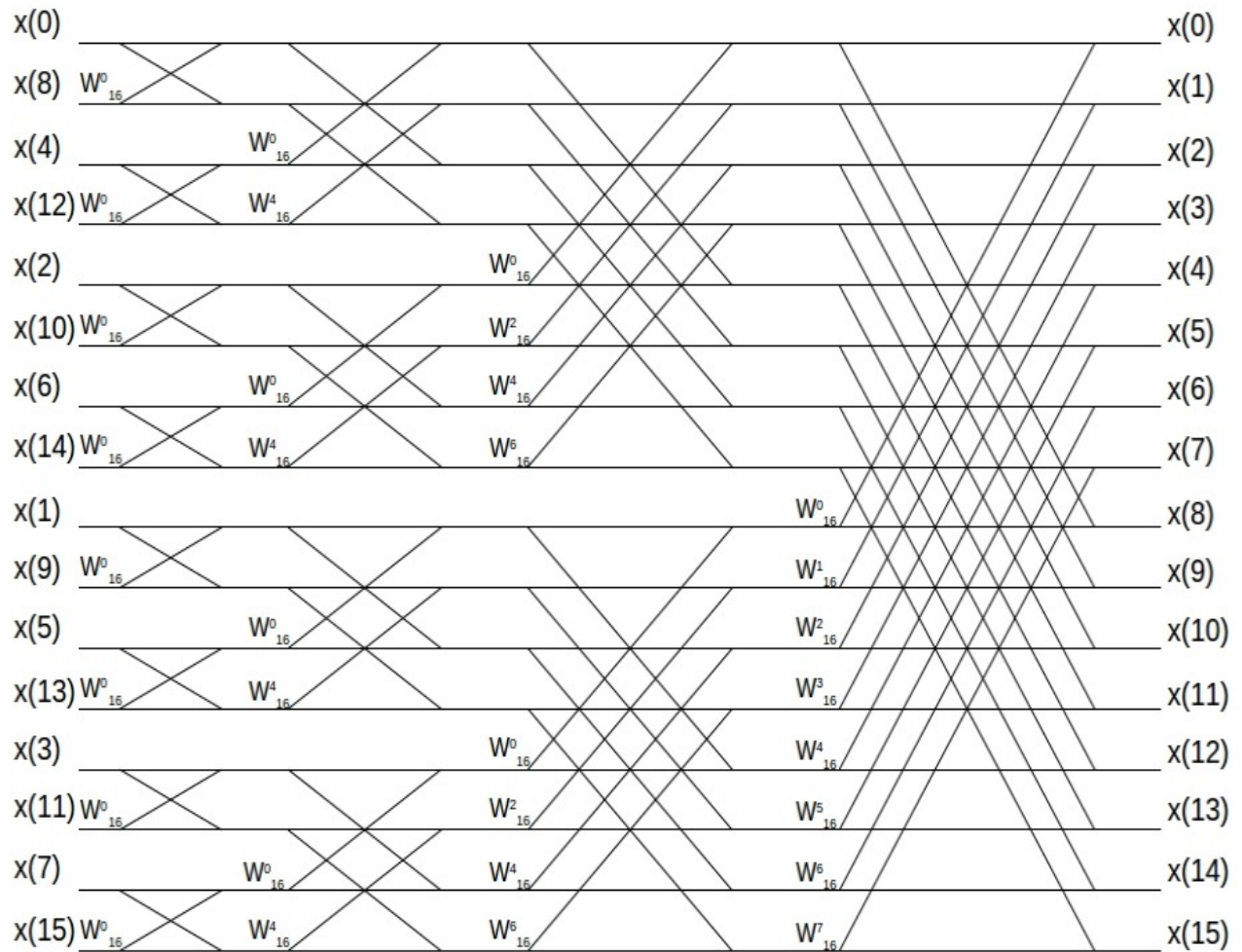


Όπως φαίνεται και από το σχήμα η κάθε πεταλούδα χρειάζεται δύο μιγαδικούς πολλαπλασιασμούς και δύο μιγαδικές αθροίσεις. Όμως ισχύει $W^{r+N/2}=W^r e^{-i\pi}=-W^r$. Με αυτόν τον μετασχηματισμό η πεταλούδα χρησιμοποιεί ένα πολλαπλασιασμό αντί για δύο. Μόνο η δεύτερη είσοδος πολλαπλασιάζεται με την σταθερά και το παράγωγο της προστίθεται με την πρώτη είσοδο (αυτή είναι η πρώτη έξοδος) και αφαιρείται από την πρώτη είσοδο (δεύτερη έξοδος).



Όπως φαίνεται η πεταλούδα αυτή είναι “φτηνότερη” από την προηγούμενη λόγω του ότι χρειάζεται έναν πολλαπλασιασμό λιγότερο. Δηλαδή χρειάζεται λιγότερες πράξεις, που σημαίνει ότι το κύκλωμα που θα υλοποιηθεί θα είναι μικρότερο.

Η μορφή για έναν υπολογισμό 16 σημείων αναλυμένη σε πεταλούδες είναι η εξής

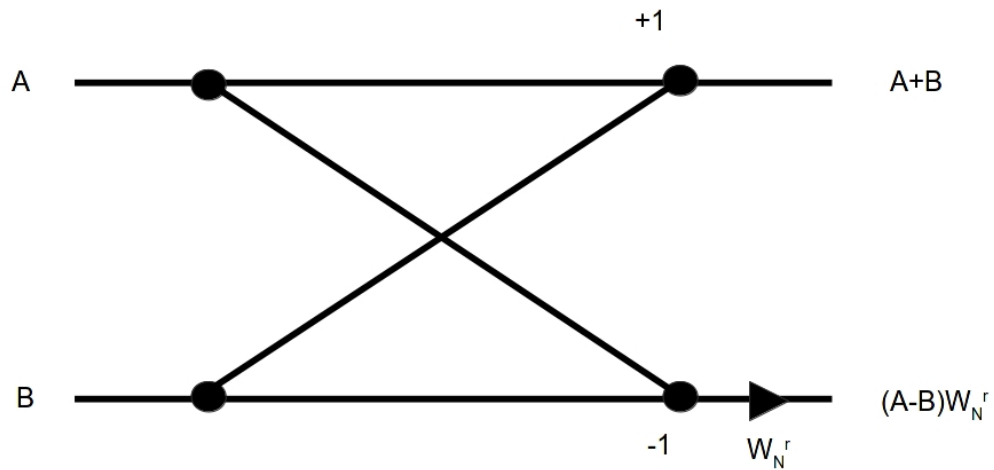


4.3.2. ΔΙΑΙΡΕΣΗ ΣΤΗ ΣΥΧΝΟΤΗΤΑ Decimation In Frequency (DIF)

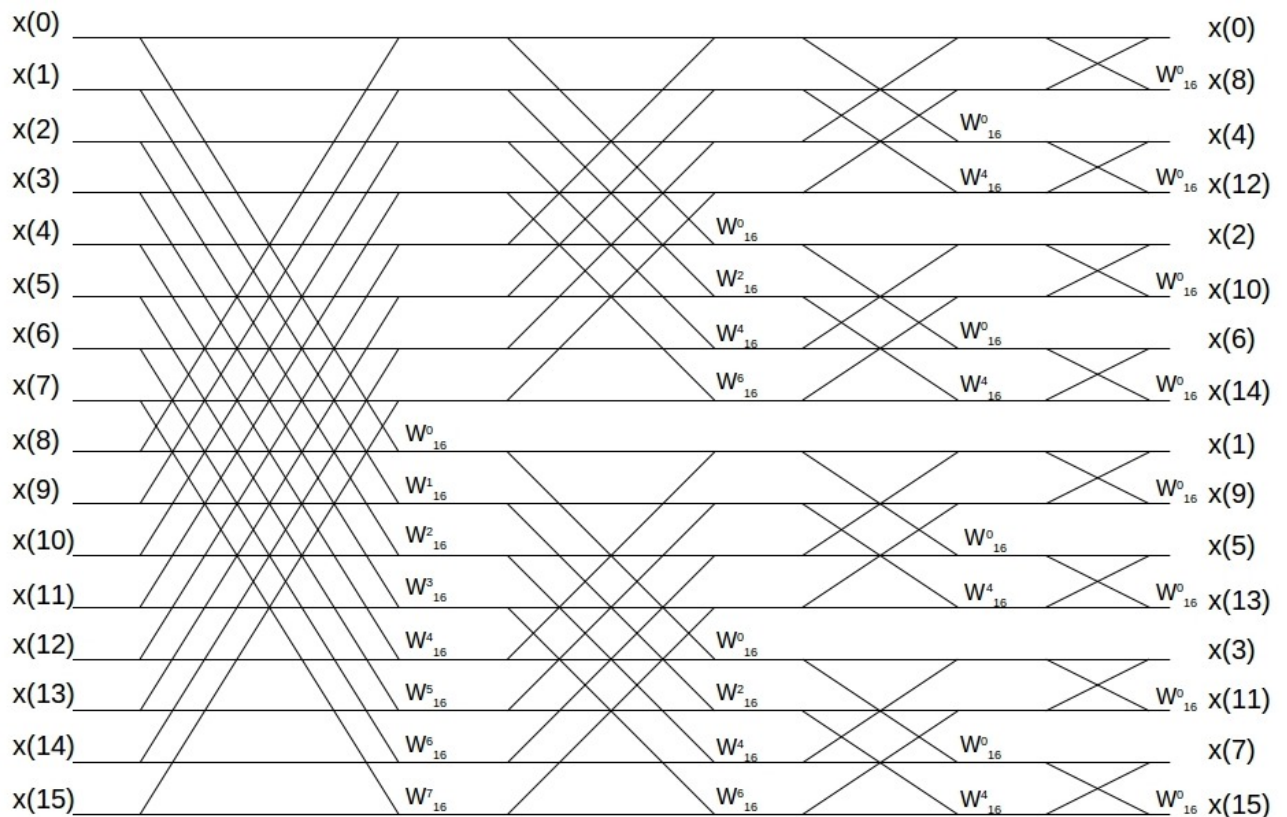
Στην προηγούμενη ανάλυση του DFT αναλύθηκε ο χρόνος (n) σε άρτιους ($2l$) και περιττούς ($2l+1$). Μία άλλη προσέγγιση είναι να αναλυθεί σε άρτιους και περιττούς η συχνότητα (k). Αρχικά χωρίζουμε το άθροισμα σε δύο κομμάτια 0 έως $N/2 - 1$ και $N/2$ έως $N - 1$ και θέτουμε το k σε $2r$ και $2r + 1$ για να χωριστεί στο άρτιο και το περιττό κομμάτι αντίστοιχα.

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn} \\
 X(k) &= \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) W_N^{nk} \\
 X(k) &= \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{n=0}^{N/2-1} x(n+N/2) W_N^{k(n+N/2)} \\
 X(k) &= \sum_{n=0}^{N/2-1} [x(n) + x(n+N/2)(-1)^k] W_N^{nk} \\
 X(2r) &= \sum_{n=0}^{N/2-1} [x(n) + x(n+N/2)] W_{N/2}^{nr} \\
 X(2r+1) &= \sum_{n=0}^{N/2-1} [x(n) - x(n+N/2)] W_N^n W_{N/2}^{nr}
 \end{aligned}$$

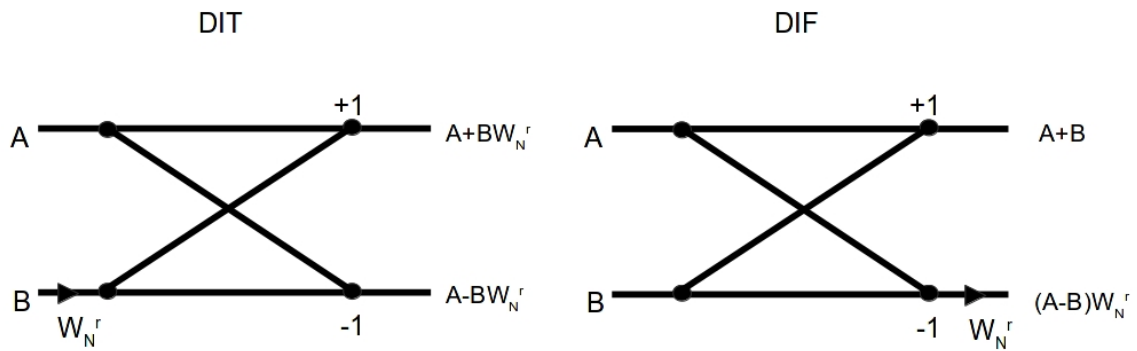
Το αποτέλεσμα είναι παρόμοιο με αυτό της προηγούμενης ανάλυσης, πάλι διακρίνονται οι μικρότεροι DFTs. Στο επίπεδο όμως της πεταλούδας (ανά δύο σημεία δηλαδή) μπορεί να παρατηρηθεί ότι η σταθερά W πολλαπλασιάζεται με την διαφορά των δύο σημείων και όχι με το δεύτερο σημείο όπως πριν. Η πεταλούδα έχει αυτήν την μορφή.



Αντίστοιχα η μορφή για έναν υπολογισμό 16 σημείων αναλυμένη σε πεταλούδες είναι η εξής



Μπορεί να παρατηρηθεί ότι η θεμελιώδης διαφορά ανάμεσα στους δύο τρόπους (DIT και DIF) είναι η μορφή της πεταλούδας



5. ΘΕΩΡΗΤΙΚΗ ΑΝΑΛΥΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ

Το κύκλωμα αυτό υλοποιεί έναν in-place radix-2 FFT DIF, έχει σχεδιαστεί έτσι ώστε να χρειάζεται όσο το δυνατόν λιγότερα transistors, να είναι δηλαδή όσο το δυνατόν μικρότερο. Σαν “in-place” εννοείται ότι ο FFT θα χρησιμοποιεί την ίδια μνήμη που χρησιμοποιεί το σήμα εισόδου, δεν χρησιμοποιεί παραπάνω μνήμη, όλος ο υπολογισμός γίνεται μέσα στην μνήμη του αρχικού σήματος. Σαν “radix-2” εννοείται ότι οι πεταλούδες δέχονται σαν είσοδο (και συνεπώς έξοδο) δύο σημεία, δηλαδή οι μικρότεροι DFTs θα είναι τεσσάρων σημείων. Επίσης για βέλτιστη απόδοση το μέγεθος του σήματος εισόδου πρέπει να είναι δύναμη του 2. Σύμφωνα με τα παραπάνω το κύκλωμα θα βασίζεται σε μια μνήμη ανάλογη της εισόδου και μια πεταλούδα δύο σημείων.

Η κάθε θέση μνήμης περιέχει δύο στοιχεία του FFT, δηλαδή χωρίζεται σε δύο μέρη. Στο παρακάτω πίνακα φαίνεται η λογική αυτή, αριστερά από κάθε κουτί είναι η διεύθυνση της θέσης. Μέσα είναι τα δύο στοιχεία με το πάνω να συμβολίζει το αριστερό κομμάτι της δυαδικής λέξης που περιέχεται στην θέση αυτήν και το κάτω το δεξί κομμάτι της. Το κάθε κομμάτι με την σειρά του αποτελείται από δύο επιμέρους κομμάτια, το ένα είναι το πραγματικό μέρος (α) και το άλλο το φανταστικό (β).

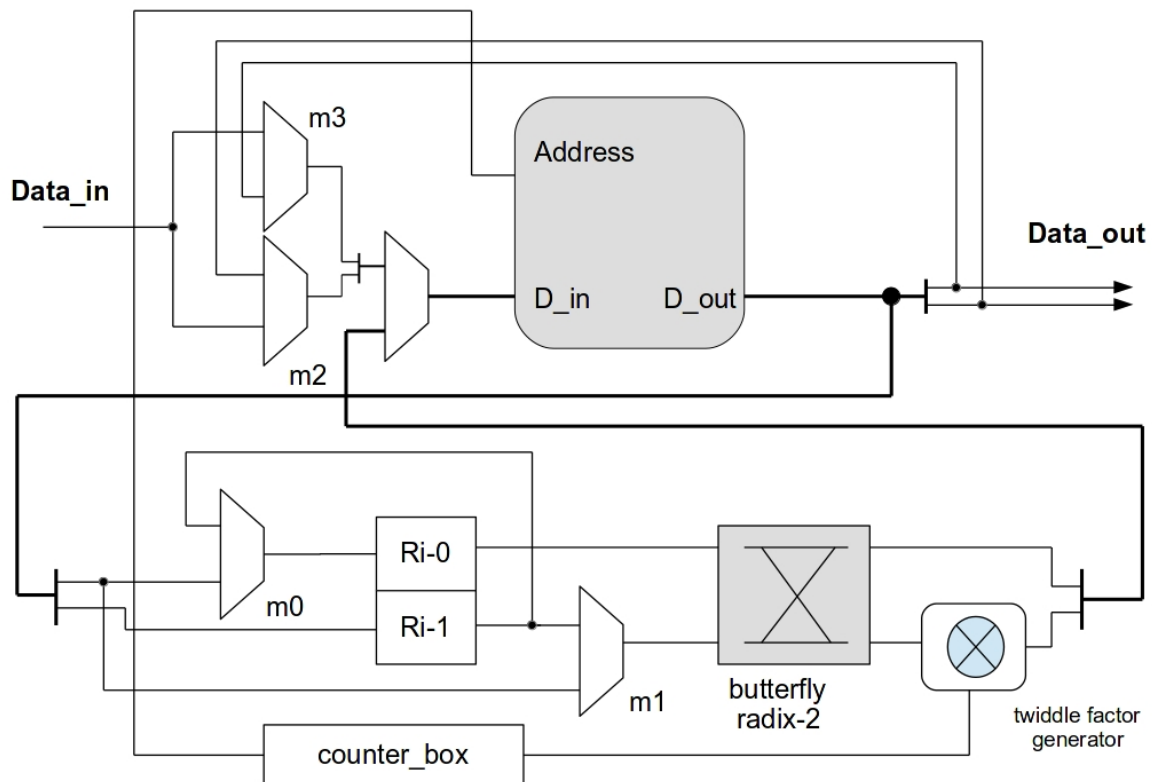
address	content
0	Re(0a) Im(0a)
	Re(0b) Im(0b)
1	Re(1a) Im(1a)
	Re(1b) Im(1b)
2	Re(2a) Im(2a)
	Re(2b) Im(2b)
3	Re(3a) Im(3a)
	Re(3b) Im(3b)
4	Re(4a) Im(4a)
	Re(4b) Im(4b)
5	Re(5a) Im(5a)
	Re(5b) Im(5b)
6	Re(6a) Im(6a)
	Re(6b) Im(6b)
7	Re(7a) Im(7a)
	Re(7b) Im(7b)

Για παράδειγμα, η θέση 3 περιέχει τα στοιχεία 3a και 3b που είναι το αριστερό και δεξί κομμάτι αντίστοιχα. Με τη σειρά του το 3a αποτελείται από τα $\text{Re}(3a)$ $\text{Im}(3a)$ που είναι το πραγματικό και φανταστικό μέρος του 3a. Έτσι η θέση 3 περιέχει την λέξη $\text{Re}(3a)$ $\text{Im}(3a)$ $\text{Re}(3b)$ $\text{Im}(3b)$.

Τα N στοιχεία εισόδου αποθηκεύονται με τη μορφή $[0 \ d_{n-2} \ d_{n-3} \ \dots \ d_1 \ d_0] \ \& \ [1 \ d_{n-2} \ d_{n-3} \ \dots \ d_1 \ d_0]$ (όπου το $\&$ συμβολίζει την ένωση των δύο λέξεων και το $n = \log_2 N$ και η λέξεις συμβολίζουν τους δείκτες των στοιχείων), δηλαδή διαφέρουν μόνο στο στοιχείο d_{n-1} , και αποθηκεύονται στην διεύθυνση $[d_{n-2} \ d_{n-3} \ \dots \ d_1 \ d_0]$. Προφανώς οι θέσεις που χρειάζονται είναι $N/2$ και οι διευθύνσεις είναι $\log_2 N - 1$ bits. Κάθε θέση μνήμης περιέχει ένα ζευγάρι από στοιχεία που είναι οι είσοδοι των πεταλούδων για το πρώτο επίπεδο (stage) του FFT. Κάθε φορά που υλοποιείται ένας υπολογισμός (πεταλούδα) τα αποτελέσματα αποθηκεύονται στις θέσεις από τις οποίες διαβάστηκαν. Αυτό όμως δεν ισχύει για τα υπόλοιπα stages. Έστω ο ακέραιος k (θα εξηγηθεί πιο κάτω το πεδίο τιμών του) οι θέσεις A_k και $A_{k+N/2^i}$ (όπου i είναι το stage) περιέχουν τα στοιχεία D_k & $D_{k+N/2^i}$ και $D_{k+2N/2^i}$ & $D_{k+3N/2^i}$, όπου το D_k και $D_{k+2N/2^i}$ είναι ένα ζευγάρι προς υπολογισμό (είσοδος πεταλούδας), ενώ το $D_{k+N/2^i}$ και $D_{k+3N/2^i}$ είναι πάλι ένα ζευγάρι προς υπολογισμό. Αφού γίνουν οι υπολογισμοί το πρώτο ζεύγος θα αποθηκευτεί στη πρώτη θέση (A_k) και το δεύτερο στην δεύτερη ($A_{k+N/2^i}$). Έτσι στις θέσεις καταλήγουν να έχουν αλλάξει θέση το $D_{k+N/2^i}$ με το $D_{k+2N/2^i}$. Οι διευθύνσεις αυτές διαφέρουν μόνο στο $n-1-i$ bit. Όταν γίνουν όλοι οι υπολογισμοί ολοκληρώνεται το stage και μετά από όλα τα stages έχει ολοκληρωθεί ο FFT.

6. ΥΛΟΠΟΙΗΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ

Το κύκλωμα που αναπτύχθηκε είναι το εξής



Αποτελείται από το εξής κομμάτια:

- (1) λογική εισόδου που αποτελείται από τρεις πολυπλέκτες
- (2) μνήμη για τους υπολογισμούς είναι μια μνήμη RAM
- (3) λογική επιλογής των δύο ζευγαριών που αποτελείται από δύο πολυπλέκτες και δύο καταχωρητές
- (4) λογική διαχείρισης του κυκλώματος (control box)
- (5) επεξεργαστής υπολογισμών (butterfly radix-2)
- (6) διαχείριση σταθερών W_N (twiddle factor generator)

(1) Λογική εισόδου

Είναι διακριτές τρεις περιπτώσεις/καταστάσεις στις οποίες μπορεί να βρεθεί το υπό-κύκλωμα αυτό.

Πρώτων να λειτουργεί έτσι ώστε να προωθεί την είσοδο του κυκλώματος στην μνήμη, έτσι αποθηκεύεται το αρχικό σήμα. Δεύτερον να προωθεί την έξοδο της πεταλούδας στην μνήμη, έτσι αποθηκεύονται οι τιμές των ενδιάμεσων στοιχείων ώστε να μπορούν να γίνουν οι επόμενοι.

Τρίτων δέχεται σαν είσοδο την έξοδο της μνήμης.

(2) Μνήμη RAM

Είναι μια μνήμη η οποία ακολουθεί την λογική που αναφέρθηκε παραπάνω. Έχει μέγεθος $N/2$ και η κάθε θέση έχει μέγεθος $2n$ επειδή περιέχει δύο στοιχεία του FFT. Οι διευθύνσεις των θέσεων έχουν $n-1$ bits, επειδή οι θέσεις είναι $N/2$ και όχι N .

(3) Λογική ζευγαριών

Όπως αναφέρθηκε παραπάνω το υπό-κύκλωμα αποτελείται από δύο καταχωρητές $Ri-0$ $Ri-1$ και δύο πολυπλέκτες $m0$ $m1$. Στο πρώτο stage τα ζεύγη D_k , $D_{k+N/2}$ από την διεύθυνση A_k , προωθούνται στους $Ri-0$ και $Ri-1$ αντίστοιχα, επειδή η κάθε θέση περιέχει τα ζευγάρια των πεταλούδων. Οι καταχωρητές προωθούν τα δεδομένα στον επεξεργαστή. Στα επόμενα stages δουλεύει ως εξής :

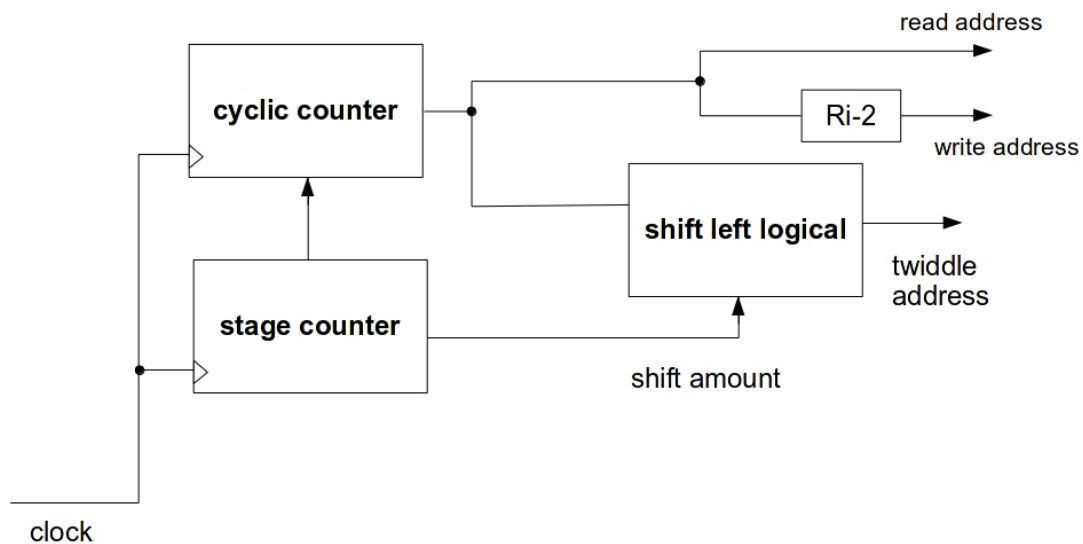
Στον πρώτο κύκλο διαβάζεται από την μνήμη το ζευγάρι D_k , $D_{k+N/2}$ από την διεύθυνση A_k και αποθηκεύονται στους καταχωρητές $Ri-0$ και $Ri-1$ αντίστοιχα.

Στον δεύτερο κύκλο διαβάζεται από την μνήμη το ζευγάρι $D_{k+2N/2}$, $D_{k+3N/2}$ από την διεύθυνση $A_{k+N/2}$. Το $D_{k+2N/2}$ προωθείται μέσω του $m1$ στον επεξεργαστή μαζί με το περιεχόμενο του $Ri-0$ (το οποίο είναι το D_k), έτσι η είσοδος του επεξεργαστή είναι το σωστό ζευγάρι (D_k , $D_{k+2N/2}$). Επίσης το περιεχόμενο του $Ri-1$ (που είναι το $D_{k+N/2}$) γράφεται στον $Ri-0$ μέσω του $m0$, ενώ το $D_{k+3N/2}$ γράφεται στον $Ri-1$.

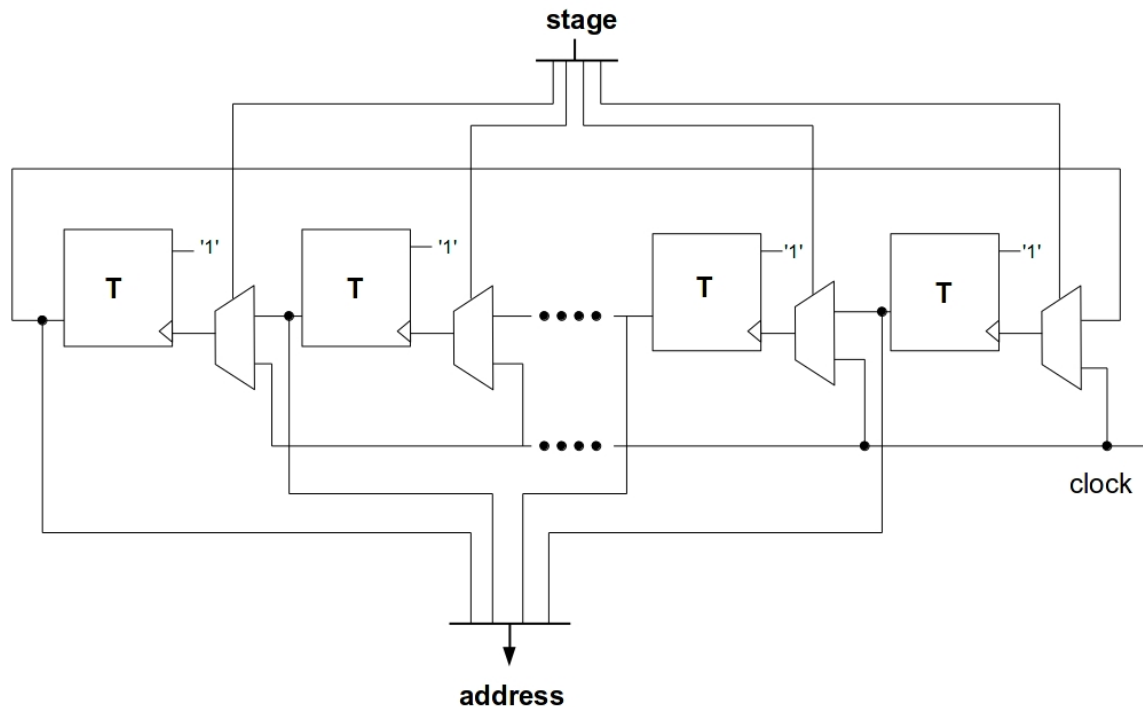
Στον τρίτο κύκλο προωθούνται τα περιεχόμενα των καταχωρητών στον επεξεργαστή. Τα περιεχόμενα είναι το ζευγάρι ($D_{k+N/2}$, $D_{k+3N/2}$).

(4) Control box

Το υπό-κύκλωμα έχει αυτή την λογική



Ο cyclic counter είναι ένας κυκλικός μετρητής που παράγει τις διευθύνσεις. Στο πρώτο stage μετράει κανονικά σαν απλός μετρητής, στα υπόλοιπα θεωρεί σαν least significant bit (lsb) το $n-i-1$ (i είναι το stage μετρούμενο από 0 έως $\log_2 N - 1$ και το -1 επειδή τα bit είναι $n-1$) ακολουθεί το σχέδιο του και ένα παράδειγμα για $N=16$.



Stage 1	Stage 2	Stage 3	Stage 4
0 0 0	0 0 0	0 0 0	0 0 0
0 0 1	1 0 0	0 1 0	0 0 1
0 1 0	0 0 1	0 0 1	0 1 0
0 1 1	1 0 1	0 1 1	0 1 1
1 0 0	0 1 0	1 0 0	1 0 0
1 0 1	1 1 0	1 1 0	1 0 1
1 1 0	0 1 1	1 0 1	1 1 0
1 1 1	1 1 1	1 1 1	1 1 1
lsb lsb		lsb	lsb

Ο stage counter είναι ένας απλός μετρητής που μετράει τα stages μεγέθους $\log_2 \log_2 N$ (τον μέγιστο ακέραιο που προκύπτει). Η τιμή του έχει δύο χρησιμότητες. Πρώτον καθορίζει πως μετράει ο cyclic counter, ορίζει δηλαδή ποιο είναι το least significant bit, δεύτερον υπολογίζει την διεύθυνση της σταθεράς W_N (twiddle) που θα χρησιμοποιηθεί στον εκάστοτε υπολογισμό. Αυτό επιτυγχάνεται ως εξής: εάν στην διεύθυνση των δεδομένων γίνει λογική ολίσθηση προς τα αριστερά (shift left logical - sll()) κατά την τιμή του stage, η παραγόμενη τιμή είναι η διεύθυνση της σταθεράς. Ακολουθεί παράδειγμα για $N=16$.

Stage 1		Stage 2	
address	twiddle	address	twiddle
0 0 0	0 0 0	0 0 0	0 0 0
0 0 1	0 0 1	1 0 0	0 0 0
0 1 0	0 1 0	0 0 1	0 1 0
0 1 1	SII(0) 0 1 1	1 0 1	SII(1) 0 1 0
1 0 0	1 0 0	0 1 0	1 0 0
1 0 1	1 0 1	1 1 0	1 0 0
1 1 0	1 1 0	0 1 1	1 1 0
1 1 1	1 1 1	1 1 1	1 1 0

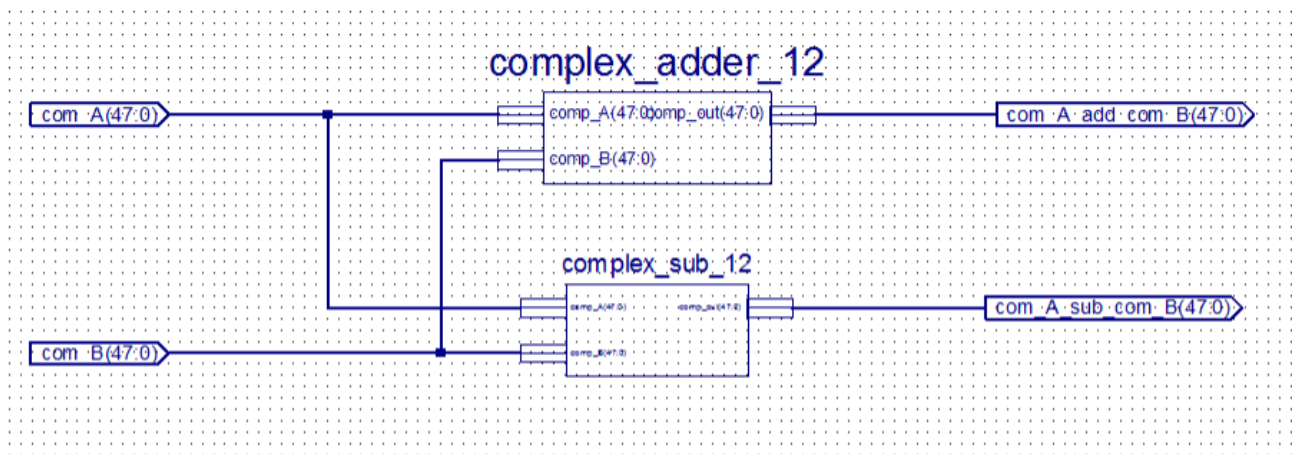
Stage 3		Stage 4	
address	twiddle	address	twiddle
0 0 0	0 0 0	0 0 0	0 0 0
0 1 0	0 0 0	0 1 0	0 0 0
0 0 1	1 0 0	0 0 1	0 0 0
0 1 1	SII(2) 1 0 0	0 1 1	SII(3) 0 0 0
1 0 0	0 0 0	1 0 0	0 0 0
1 1 0	0 0 0	1 1 0	0 0 0
1 0 1	1 0 0	1 0 1	0 0 0
1 1 1	1 0 0	1 1 1	0 0 0

Σύμφωνα με την προηγούμενη ενότητα (3) ένα ζευγάρι υπολογισμών διαρκεί τρεις κύκλους. Στον πρώτο κύκλο διαβάζονται δεδομένα από την διεύθυνση A_k . Στον δεύτερο αποθηκεύονται δεδομένα στην A_k και διαβάζονται από την $A_{k+N/2^i}$. Τέλος στον τρίτο αποθηκεύονται δεδομένα στην $A_{k+N/2^i}$. Κάθε φορά χρειάζεται καινούρια διεύθυνση για διάβασμα και η προηγούμενη για γράψιμο. Αυτό επιτυγχάνεται με τον καταχωρητή Ri-2 στην έξοδο του cyclic counter και μια έξοδο χωρίς καταχωρητή.

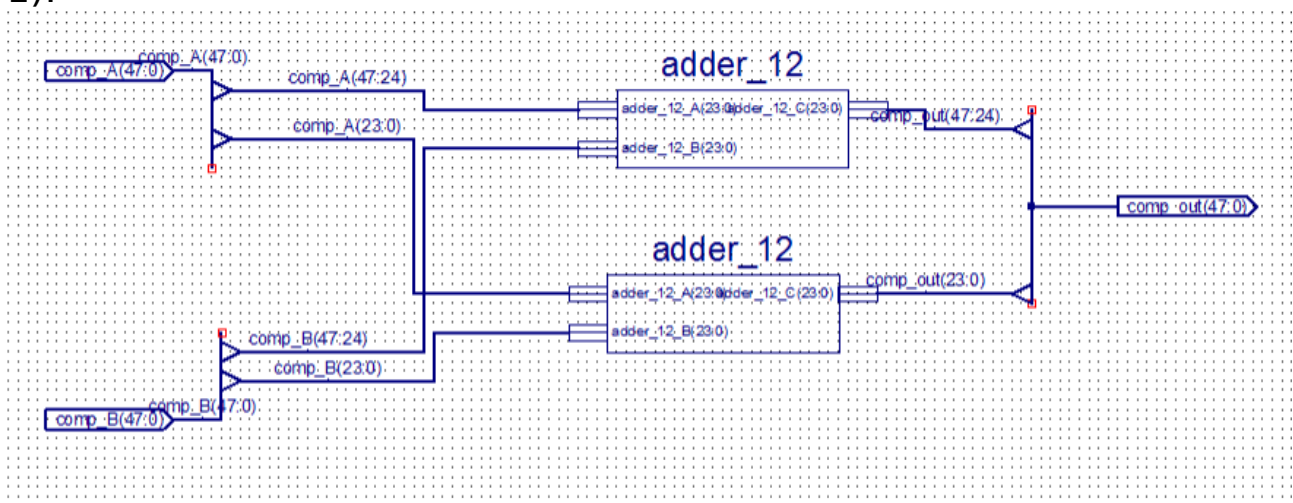
Επίσης έχει υλοποιηθεί ένα FSM το οποίο διαχειρίζεται τα σήματα ελέγχου για την μνήμη και τους πολυπλέκτες ανάλογα με τις τιμές του cyclic counter και του stage counter.

(5) Butterfly

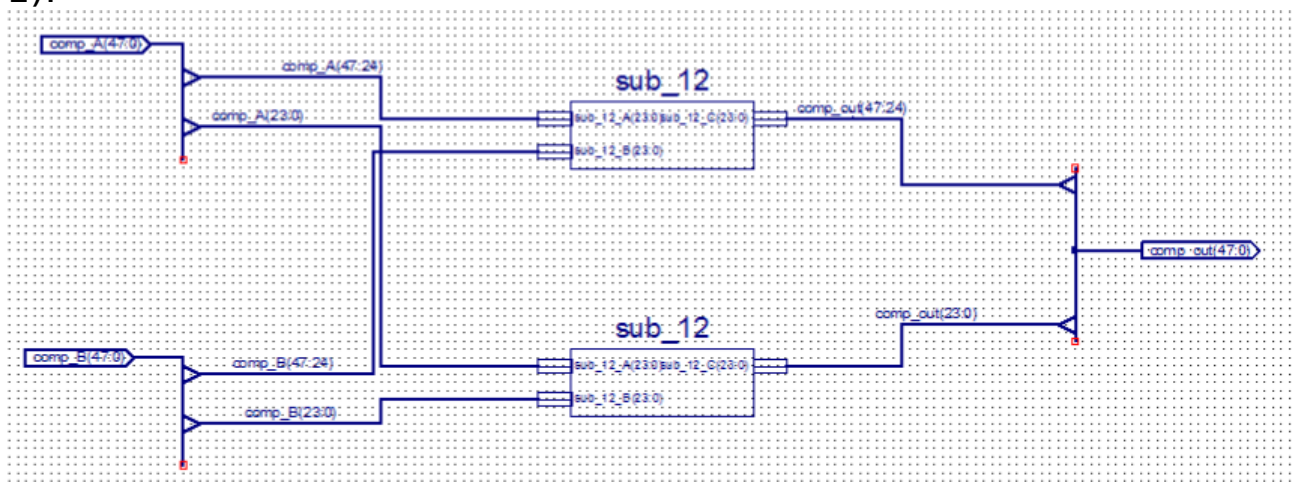
Ο επεξεργαστής αποτελείται από έναν μιγαδικό αθροιστή και έναν μιγαδικό αφαιρέτη (κεφάλαιο 1). Έχει δύο εισόδους και δύο εξόδους. Η μια έξοδος παράγει το άθροισμα των εισόδων και η άλλη την διαφορά.



Ο μιγαδικός αθροιστής αποτελείται από δύο αθροιστές (κεφάλαιο 1).



Ο μιγαδικός αφαιρέτης αποτελείται από δύο αφαιρέτες (κεφάλαιο 1).

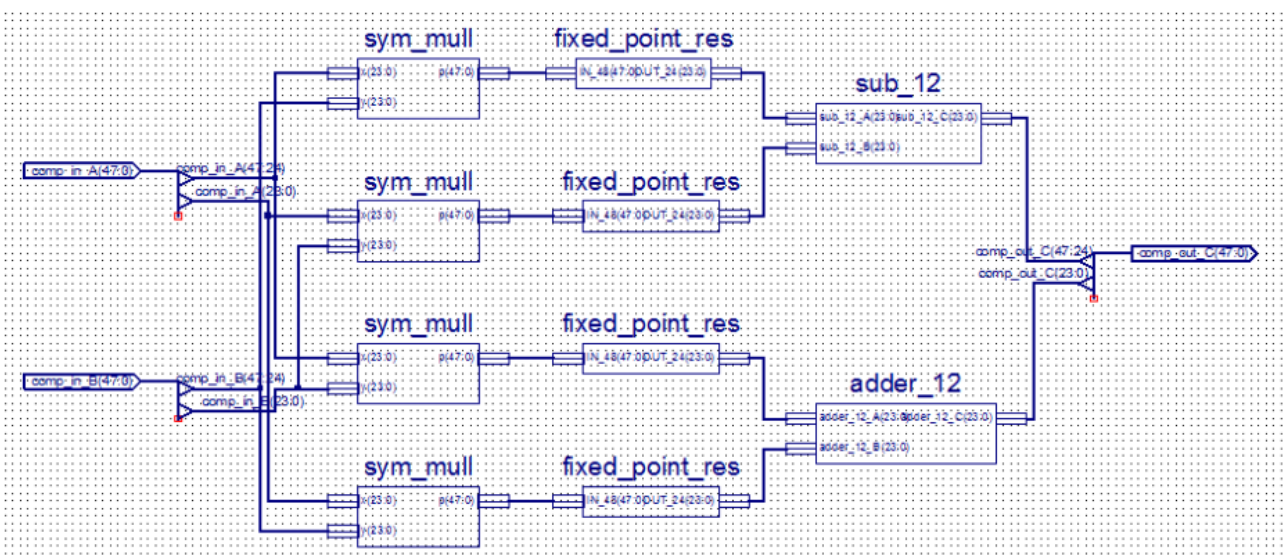
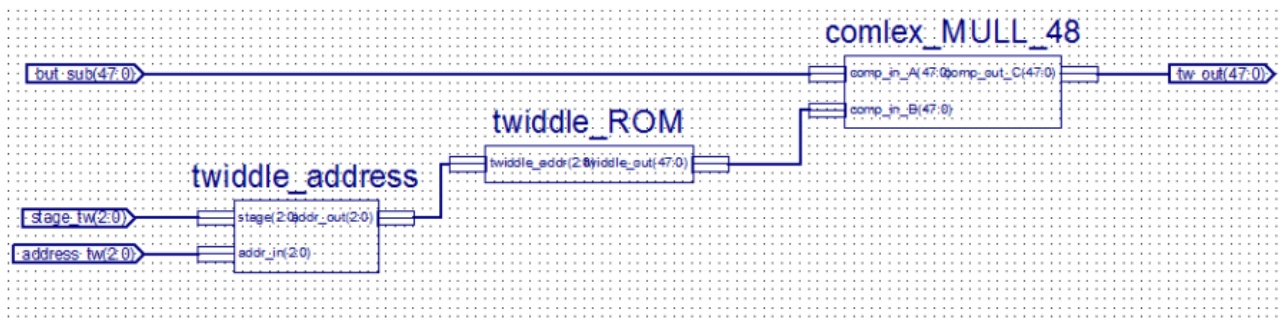


Από τα παραπάνω σχήματα επίσης φαίνεται πως αναπαριστάται ο μιγαδικός αριθμός σε καλώδιο/δυαδική λέξη. Τα πρώτα από

αριστερά bit (47 έως 24) είναι το πραγματικό μέρος (α) και υπόλοιπα (23 έως 0) το φανταστικό μέρος (β). Το κάθε μέρος και από τους δύο μιγαδικούς αριθμούς δρομολογείται στον αντίστοιχο αθροιστή ή αφαιρέτη, με αποτέλεσμα να γίνει η πράξη ανάμεσα στα πραγματικά μέρη και τα φανταστικά.

(6) twiddle factor generator

Στο υπό-κύκλωμα αυτό υπάρχει μια μνήμη ROM που έχει αποθηκευμένα τα twiddles, σε κάθε θέση μνήμης βρίσκεται το αντίστοιχο twiddle. Τα twiddles είναι $N/2$ σε πλήθος και είναι οι ρίζες του μοναδιαίου μιγαδικού αριθμού W_N (κεφάλαιο 1). Επίσης έχει υλοποιηθεί και ένας μιγαδικός πολλαπλασιαστής (κεφάλαιο 1) που παράγει το αποτέλεσμα του twiddle με την δεύτερη έξοδο του επεξεργαστή (η έξοδος που προκύπτει από την διαφορά των εισόδων του επεξεργαστή).



Οι πολλαπλασιαστές έχουν υλοποιηθεί με array multipliers που μπορούν να διαχειριστούν αριθμούς σε συμπλήρωμα-2 (2's complement).

Με αυτήν την υλοποίηση μπορεί το κύκλωμα να διαχειριστεί την επέκταση του πρόσημου (sign extent), και έτσι το αποτέλεσμα της πράξης είναι σωστό. Με λίγα λόγια μιμείται την επιμεριστική ιδιότητα ακριβώς όπως θα γινόταν αν η πράξη γινόταν στο χαρτί, με επέκταση πρόσημου προφανώς.

Το component fixed_point_res είναι απλά ένας δίαυλος (bus) που κρατάει μόνο τα ψηφία που χρειάζονται από τις εξόδους των πολλαπλασιαστών (όπως αναλύθηκε στην θεωρία της fixed point αριθμητικής) ώστε να διατηρηθεί η ίδια αναπαράσταση με την είσοδο.

7. ΠΑΡΑΔΕΙΓΜΑ ΛΕΙΤΟΥΡΓΕΙΑΣ

Στο πρώτο stage τα σημεία του FFT είναι αποθηκευμένα στην μνήμη έτσι ώστε κάθε θέση μνήμης να περιέχει ένα ζευγάρι, δηλαδή δύο σημεία μιας πεταλούδας. Ο μετρητής μετράει με least-significant bit το δεξί, δηλαδή με βήμα ένα και το αποτέλεσμα δίνει και την διεύθυνση του αντίστοιχου twiddle. Υπολογίζεται το κάθε ζευγάρι και αποθηκεύεται στην ίδια θέση μνήμης από την οποία διαβάστηκε.

Stage 1																		
address			index			twiddle		address-pairs			Butterfly-pairs							
0	0	0	0	0	0	0	0	0	0	-	-	-	0	0				
			1	0	0	0	8	0	0	0				1	0	0	0	8
0	0	1	0	0	0	1	1	0	0	1	-	-	-	0	0	0	1	1
			1	0	0	1	9	0	0	1				1	0	0	1	9
0	1	0	0	0	1	0	2	0	1	0	-	-	-	0	0	1	0	2
			1	0	1	0	10	0	1	0				1	0	1	0	10
0	1	1	0	0	1	1	3	0	1	1	-	-	-	0	0	1	1	3
			1	0	1	1	11	0	1	1				1	0	1	1	11
1	0	0	0	1	0	0	4	1	0	0	-	-	-	0	1	0	0	4
			1	1	0	0	12	1	0	0				1	1	0	0	8
1	0	1	0	1	0	1	5	1	0	1	-	-	-	0	1	0	1	5
			1	1	0	1	13	1	0	1				1	1	0	1	13
1	1	0	0	1	1	0	6	1	1	0	-	-	-	0	1	1	0	6
			1	1	1	0	14	1	1	0				1	1	1	0	14
1	1	1	0	1	1	1	7	1	1	1	-	-	-	0	1	1	1	7
			1	1	1	1	15	1	1	1				1	1	1	1	15

Κατά το δεύτερο stage ο μετρητής μετράει με least-significant bit το αριστερό, ανά τέσσερα. Έτσι ανά δύο διευθύνσεις προκύπτουν τα δύο ζεύγους που θα πρέπει να υπολογιστούν. Οι διευθύνσεις είναι οι (0,4) (1,5) (2,6) (3,7) και τα αντίστοιχα twiddles, που προκύπτουν με shift left κατά ένα, είναι (0,0) (2,2) (4,4) (6,6). Το ένα ζεύγος των σημείων σε κάθε περίπτωση είναι τα αριστερά μέρη του κάθε ζεύγους διευθύνσεων και αντίστοιχα το

άλλο ζεύγος σημείων είναι τα δεξιά μέλη των θέσεων μνήμης. Στην περίπτωση του (0,4) τα στοιχεία έχουν την μορφή ([0 8],[4 12]) θα γίνουν οι υπολογισμοί μεταξύ των 0 - 4 και 8 - 12 και θα αποθηκευτούν στις αντίστοιχες θέσεις με την σειρά, καταλήγουν δηλαδή στην μορφή ([0 4],[8 12]) στις διευθύνσεις (0,4).

Stage 2

address	index	twiddle	address-pairs	Butterfly-pairs
0 0 0	0 0 0 0	0	0 0 0	0 0 0 0 0
	1 0 0 0			0 1 0 0 4
0 0 1	0 0 0 1	1	1 0 0	1 0 0 0 8
	1 0 0 1	9		1 1 0 0 12
0 1 0	0 0 1 0	2	0 0 1	0 0 0 1 1
	1 0 1 0	10		0 1 0 1 5
0 1 1	0 0 1 1	3	1 0 1	1 0 0 1 9
	1 0 1 1	11		1 1 0 1 13
1 0 0	0 1 0 0	4	0 1 0	0 0 1 0 2
	1 1 0 0	8		0 1 1 0 6
1 0 1	0 1 0 1	5	1 1 0	1 0 1 0 10
	1 1 0 1	13		1 1 1 0 14
1 1 0	0 1 1 0	6	0 1 1	0 0 1 1 3
	1 1 1 0	14		0 1 1 1 7
1 1 1	0 1 1 1	7	1 1 1	1 0 1 1 11
	1 1 1 1	15		1 1 1 1 15

Στο τρίτο stage ο μετρητής μετράει με least-significant bit το μεσαίο, ανά δύο. Οι διευθύνσεις που προκύπτουν είναι (0,2) (1,3) (4,6) (5,7) και τα αντίστοιχα twiddles, που προκύπτουν με shift left κατά δύο, είναι (0,0) (4,4) (0,0) (4,4). Έστω το (4,6) που έχει τη μορφή ([8 12],[10 14]) θα γίνουν οι υπολογισμοί μεταξύ των 8 - 10 και 12 - 14 και θα αποθηκευτούν στις αντίστοιχες θέσεις με την σειρά, καταλήγουν δηλαδή στην μορφή ([8 10],[12 14]) στις διευθύνσεις (4,6).

Stage 3

address	index		twiddle	address-pairs	Butterfly-pairs
0 0 0	0 0 0 0	0	0 0 0	0 0 0	0 0 0 0 0
	0 1 0 0	4			0 0 1 0 2
0 0 1	0 0 0 1	1	1 0 0	0 1 0	0 1 0 0 4
	0 1 0 1	5			0 1 1 0 6
0 1 0	0 0 1 0	2	0 0 0	0 0 1	0 0 0 1 1
	0 1 1 0	6			0 0 1 1 3
0 1 1	0 0 1 1	3	1 0 0	0 1 1	0 1 0 1 5
	0 1 1 1	7			0 1 1 1 7
1 0 0	1 0 0 0	8	0 0 0	1 0 0	1 0 0 0 8
	1 1 0 0	12			1 0 1 0 10
1 0 1	1 0 0 1	9	1 0 0	1 1 0	1 1 0 0 12
	1 1 0 1	13			1 1 1 0 14
1 1 0	1 0 1 0	10	0 0 0	1 0 1	1 0 0 1 9
	1 1 1 0	14			1 0 1 1 11
1 1 1	1 0 1 1	11	1 0 0	1 1 1	1 1 0 1 13
	1 1 1 1	15			1 1 1 1 15

Στο τέταρτο stage ο μετρητής μετράει με least-significant bit το αριστερό, ανά ένα. Οι διευθύνσεις που προκύπτουν είναι (0,1) (2,3) (4,5) (6,7) και τα αντίστοιχα twiddles, που προκύπτουν με shift left κατά τρία, είναι (0,0) (0,0) (0,0) (0,0). Έστω το (2,3) που έχει τη μορφή ([4 6],[5 7]) θα γίνουν οι υπολογισμοί μεταξύ των 4 - 5 και 6 - 7 και θα αποθηκευτούν στις αντίστοιχες θέσεις με την σειρά, καταλήγουν δηλαδή στην μορφή ([4 5],[6 7]) στις διευθύνσεις (2,3).

Stage 4

address	index		twiddle	address-pairs	Butterfly-pairs
0 0 0	0 0 0 0	0	0 0 0	0 0 0	0 0 0 0 0
	0 0 1 0	2			0 0 0 1 1
0 0 1	0 0 0 1	1	0 0 0	0 0 1	0 0 1 0 2
	0 0 1 1	3			0 0 1 1 3
0 1 0	0 1 0 0	4	0 0 0	0 1 0	0 1 0 0 4
	0 1 1 0	6			0 1 0 1 5
0 1 1	0 1 0 1	5	0 0 0	0 1 1	0 1 1 0 6
	0 1 1 1	7			0 1 1 1 7
1 0 0	1 0 0 0	8	0 0 0	1 0 0	1 0 0 0 8
	1 0 1 0	10			1 0 0 1 9
1 0 1	1 0 0 1	9	0 0 0	1 0 1	1 0 1 0 10
	1 0 1 1	11			1 0 1 1 11
1 1 0	1 1 0 0	12	0 0 0	1 1 0	1 1 0 0 12
	1 1 1 0	14			1 1 0 1 13
1 1 1	1 1 0 1	13	0 0 0	1 1 1	1 1 1 0 14
	1 1 1 1	15			1 1 1 1 15

Στην μνήμη μετά από το τελευταίο stage είναι αποθηκευμένα τα στοιχεία με την σειρά. Για παράδειγμα στην θέση 7 είναι αποθηκευμένα τα στοιχεία [14 15].

Result	
address	index
0 0 0	0 0 0 0 0 0
	0 0 0 0 1 1
0 0 1	0 0 1 0 0 2
	0 0 1 0 1 3
0 1 0	0 1 0 0 0 4
	0 1 0 0 1 5
0 1 1	0 1 1 0 0 6
	0 1 1 0 1 7
1 0 0	1 0 0 0 0 8
	1 0 0 0 1 9
1 0 1	1 0 1 0 0 10
	1 0 1 0 1 11
1 1 0	1 1 0 0 0 12
	1 1 0 0 1 13
1 1 1	1 1 1 0 0 14
	1 1 1 0 1 15

Τέλος στα στοιχεία πρέπει να γίνει αναστροφή bit όπως ορίζει ο αλγόριθμος του FFT.

8. ΑΠΟΤΕΛΕΣΜΑΤΑ

Το κύκλωμα αυτό όπως αναφέρθηκε πιο πάνω έχει σχεδιαστεί έτσι ώστε να χρειάζεται όσο το δυνατόν λιγότερα transistors το κομμάτι που αποτελείται από τις ενότητες (3) και (4), δηλαδή η διευθυνσιοδότηση-επιλογή ζευγαριών. Ακολουθεί ένας πίνακας που περιέχει πόσα transistors χρειάζεται το κάθε στοιχείο του υπό-κυκλώματος για K bits για τις διευθύνσεις και m bits για τα stages .

	AND	FF	MUX	trans
counter	K-2	K	0	18*K-12
cyclic counter	0	K	K	18*K
register	0	K	0	10*K
shift left logical by m	0	0	K*m	6*K*m
mux	0	0	1	6

Εάν εφαρμοστούν αυτά στο παρόν κύκλωμα προκύπτουν τα εξείς για N από 16 έως 32768.

	16	32	64	128	256	512
1 stage counter	24	42	42	42	42	60
1 cyclic counter	72	90	108	126	144	162
2 registers	800	800	800	800	800	800
2 MUX	480	480	480	480	480	480
address register	30	40	50	60	70	80
shifter	36	72	90	108	126	192
SUM	1442	1524	1570	1616	1662	1774
	1024	2048	4096	8192	16384	32768
1 stage counter	60	60	60	60	60	60
1 cyclic counter	180	198	216	234	252	270
2 registers	800	800	800	800	800	800
2 MUX	480	480	480	480	480	480
address register	90	100	110	120	130	140
shifter	216	240	264	288	312	336
SUM	1826	1878	1930	1982	2034	2086

Συγκριτικά με τις τεχνικές [3] και [8], οι οποίες χρησιμοποιούν ένα μεγάλο σύνολο από αναδιατάξεις και εφαρμόζει διαφορετικές αναδιατάξεις σε κάθε stage, η παρούσα τεχνική χρησιμοποιεί δύο συγκεκριμένες αναδιατάξεις (πολυπλέκτες m_0 και m_1 για την είσοδο της πεταλούδας και ο πολυπλέκτης της εισόδου της μνήμης για την έξοδο της πεταλούδας). Η καθυστέρηση του κυκλώματος της παραγωγής διευθύνσεων είναι τόση όση η καθυστέρηση ενός counter σε σχέση με την καθυστέρηση ενός counter και ενός δέντρου αθροιστών στις τεχνικές [3], [7], [12]. Ο counter και ο barrel shifter της [6] καθιστούν μια απλή τεχνική διευθυνσιοδότησης, όμως οι $2 \cdot b^2$ καταχωρητές δεν μπορούν να θεωρηθούν βελτίωση καθώς μεγαλώνει ο αριθμός των τραπεζών. Εν τέλη, η προτεινόμενη τεχνική απλοποιεί το σχέδιο επειδή χρησιμοποιεί μια τράπεζα, βελτιώνει την απαιτούμενη πολυπλοκότητα του υλισμικού για την διευθυνσιοδότηση καθώς και την καθυστέρηση σε σχέση με τις προαναφερθείσες τεχνικές.

9. ΑΝΑΦΟΡΕΣ

- [1] D. S. Parker, "Notes on Shuffle/Exchange-Type Switching Networks", IEEE Trans. on Computers, 1980.
- [2] E. H. Wold and A. M. Despain, "Pipeline and Parallel FFT Processors for VLSI Implementations", IEEE Trans. on Computers, vol. C-33, 1984.
- [3] L. G. Johnson, "Conflict Free Memory Addressing for Dedicated FFT Hardware", IEEE Trans. on Circuits and Systems - II: Analog and Digital Signal Processing, Vol. 39, No. 5, May 1992.
- [4] Y. Ma, "An Effective Memory Addressing Scheme for FFT Processors", IEEE Trans. on Signal Processing, Vol. 47, No. 3, pp. 907-911, May 1999.
- [5] X. Xiao, E. Oruklu and J. Saniie, "An Efficient FFT Engine with Reduced Addressing Logic", IEEE Trans. on Circuits and Systems II: Express Briefs, pp 1149-1153, Nov. 2008.
- [6] X. Xiao, E. Oruklu and J. Saniie, "Fast Memory Addressing Scheme for Radix-4 FFT Implementation", IEEE Intl. Conf. on Electro / Information Technology, 2009.
- [7] J. H. Takala, T. S. Järvinen, and H. T. Sorokin, "Conflict-Free Parallel Memory Access Scheme for FFT Processors" in the Proc. of the Intl. Symposium on Circuits and Systems, ISCAS 2003, Vol-4 pp 524-527.
- [8] D. Reisis, N. Vlassopoulos, "Conflict Free Parallel Memory Accessing Techniques for FFT Architectures" IEEE Trans. on Circuits and Systems I, Vol. 55, No 11, Dec 2008, pp 3438-3447.
- [9] K. Nakos, D. Reisis, N. Vlassopoulos, "Addressing Technique for Parallel Memory Accessing in Radix-2 FFT Processors" in the Proceedings of the IEEE ICECS, pp.52-56, September 2008.
- [10] M. Pease, "Organization of large scale Fourier transforms", Journal of the ACM, vol 16, pp 474-482, July 1969.
- [11] D. Cohen, "Simplified control of FFT hardware", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 577-579, Dec. 1976.
- [12] M. Püschel, P. A. Milder, J. C. Coe, "Permuting Streaming Data Using RAMs", Journal of the ACM, Vol 56, No 2. Article 10, April 2009.
- [13] Byung G. Jo and Myung H. Sunwoo, "New Continuous-Flow Mixed-Radix (CFMR) FFT Processor Using Novel In-Place Strategy", IEEE Trans. on Circuits and Systems-I:Regular Papers, vol. 52, No, 5, MAY 2005.
- [14] Anthony T. Jacobson, Dean N. Truong, and Bevan M. Baas, "The Design of a reconfigurable Continuous-Flow Mixed-Radix FFT Processor", Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium, 24-27 May 2009.