**ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ**

# Εργαστήριο 8

## Ομάδα 09 – Group 1

|  |  |  |
| --- | --- | --- |
| **ΕΠΩΝΥΜΟ** | **ΟΝΟΜΑ** | **ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ** |
| ΔΑΣΟΥΛΑΣ | ΙΩΑΝΝΗΣ | 1053711 |
| ΔΟΥΡΔΟΥΝΑΣ | ΑΡΙΣΤΕΙΔΗΣ ΑΝΑΡΓΥΡΟΣ | 1047398 |

Υλοποίηση Φασματικού Αναλυτή

**Στόχος - Εισαγωγή:**

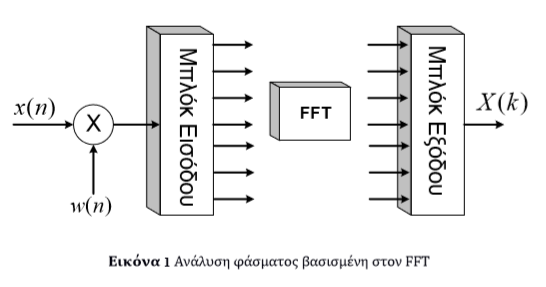
Σε αυτό το εργαστήριο θα υλοποιηθεί ο γρήγορος μετασχηματισμός Fourier (FFT), με σκοπό την φασματική ανάλυση σήματος σε πραγματικό χρόνο.

Ο FFT είναι ένας αποδοτικός αλγόριθμος για τον υπολογισμό του διακριτού μετασχηματισμού Fourier (DFT) Ν σημείων, ο οποίος μπορεί να θεωρηθεί ως δειγματοληψία του μετασχηματισμού Fourier διακριτού χρόνου (discrete-time Fourier transform, DTFT) και δίνεται ως,



Επειδή ο FFT υπολογίζει το φάσμα ενός μπλόκα δειγμάτων του σήματος (Εικόνα 1), ο υπολογισμός του θα γίνεται μία φορά για κάθε ένα μπλοκ δεδομένων που θα αποθηκεύεται. Σε αυτό το εργαστήριο θα υπολογίζετε FFT μήκους 64 δειγμάτων.

Η απότομη αποκοπή του σήματος στα άκρα του μπλοκ έχει ως αποτέλεσμα την παραμόρφωση του φάσματος που βλέπουμε χρησιμοποιώντας τον FFT σε σχέση με το πραγματικό φάσμα του σήματος. Για τον λόγο αυτό θα χρησιμοποιήσουμε κατάλληλη παραθύρωση των δεδομένων. Σε αυτό το εργαστήριο θα χρησιμοποιήσετε την παραθύρωση Hamming.



**Ασκήσεις:**

**Άσκηση 5.1 (Συμπλήρωση μηδενικών)**

1.Να τρέξει το παραπάνω πρόγραμμα χωρίς καμία αλλαγή. Θα πρέπει να παρατηρηθεί υψηλή φασματική ενέργεια στο1kHz. Ακόμα όμως και στην αναλυτική απεικόνιση δεν θα φανεί περισσότερη λεπτομέρεια του φάσματος του σήματος λόγω της δειγματοληψίας του DTFT στη συχνότητα.

2.Να αλλαχθεί η γραμμή 2 συμπληρώνοντας με μηδενικά το σήμα έτσι ώστε ο αριθμός των σημείων να γίνει 4096 (Μ=4096;). Να τρέξει ο κώδικας και να παρατηρηθεί ξανά το φάσμα. Ποια είναι η πραγματική διαφορά σε σύγκριση με το προηγούμενο;

1. Ο κώδικας και η έξοδος που δόθηκε από την Matlab για τον κώδικα της εκφώνησης είναι η ακόλουθη:

Εικόνα που περιέχει στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Παρατηρείται ότι παρόλο που η είσοδος είναι ένα ημίτονο συχνότητας 1 kHz ο FFT έχει τριγωνική μορφή στα 1000 Hz . Η μορφή αυτή οφείλεται στο γεγονός ότι το σήμα στο οποίο γίνεται ο DFT εμφανίζει στα άκρα ασυνέχεια λόγω του απότομου κοψίματος.

2. Αυξάνοντας τον αριθμό τον σημείων σε 4096 το φάσμα που λήφθηκε είναι το ακόλουθο:

Εικόνα που περιέχει κείμενο, χάρτης

Περιγραφή που δημιουργήθηκε αυτόματα

Με την αύξηση των δειγμάτων σε 4096 εμφανίζονται κυματώσεις στα αριστερά και στα δεξιά του φάσματος. Ωστόσο με την προσθήκη δειγμάτων λαμβάνεται μια καλύτερη εικόνα του FFT.

**Άσκηση 5.2 (Παραθύρωση)**

1. Να παρατηρηθεί το σχήμα του παραθύρου boxcar χρησιμοποιώντας την εντολή:

plot(boxcar(256));

2. Χρησιμοποιώντας Μ=4096 σημεία στην γραμμή 2 για συμπλήρωση με μηδενικά και την boxcar στην γραμμή 7, παρατηρήστε το φάσμα του ημιτονοειδούς σήματος 1kHz (στην πραγματικότητα το ίδιο κάνατε και πριν).

3. Να παρατηρηθεί το σχήμα του παραθύρου hamming χρησιμοποιώντας την εντολή:

plot(hamming(256));

4. Να πολλαπλασιαστούν τα δείγματα με ένα παράθυρο hamming και να σχεδιαστεί το μέτρο των συντελεστών του FFT σε λογαριθμική κλίμακα. Θα πρέπει να αλλάξει η γραμμή 7 με:

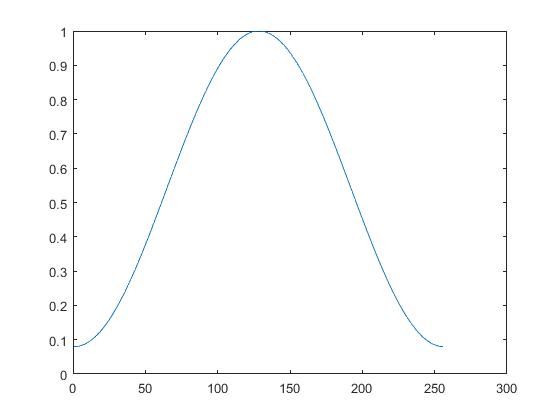
x=x.\*hamming(N)

Πώς αλλάζει το φάσμα του FFT συγκρινόμενο με την περίπτωση του boxcar παραθύρου;

5. Να περιγραφούν τα προβλήματα που δημιουργεί το boxcar παράθυρο. Ποια τα πλεονεκτήματα και ποια τα μειονεκτήματα του παραθύρου hamming ; Να περιγραφεί η σχέση μεταξύ του πλάτους του κύριου λοβού και της συμπεριφοράς των παράπλευρων λοβών των δύο παραθύρων (στη συχνότητα)

1. Εντολή plot(boxcar(256)):Εικόνα που περιέχει ουρανός, κείμενο

   Περιγραφή που δημιουργήθηκε αυτόματα
2. Η έξοδος με Μ=4096 σημεία και χρήση της εντολής boxcar είναι όμοια με αυτή των ερωτημάτων 5.1
3. Η εντολή plot(hamming(256)):



1. Ο κώδικας που αναπτύχθηκε και η έξοδος της Matlab :

Εικόνα που περιέχει στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, χάρτης

Περιγραφή που δημιουργήθηκε αυτόματα

Η χρήση του παραθύρου Hamming έχει ως αποτέλεσμα να εξαλειφθούν οι κυματώσεις στα άκρα. Αυτό επιτυγχάνεται μέσω του πολλαπλασιασμού των ακραίων σημείων με πολύ μικρές τιμές εξομαλύνοντας τις ασυνέχειες.

1. Το μειονέκτημα του boxcar παραθύρου είναι ότι πολλαπλασιάζει όλα τα δείγματα με 1 διατηρώντας έτσι τις ασυνέχειες και εμφανίζοντας στον FFT κυματώσεις σε αντίθεση με το Hamming που πολλαπλασιάζει με πολύ μικρές τιμές και εξαλείφει τις κυματώσεις. Η κατάλληλη επιλογή παραθύρου διαμορφώνει τελικά τη σχέση εύρους κεντρικού και παράπλευρων λοβών καθώς και την ζώνη μετάβασης και τις διακυμάνσεις των φίλτρων.

**Άσκηση 5.3 (Φασματικός Αναλυτής)**

Υλοποιείστε ένα φασματικό αναλυτή πραγματικού χρόνου χρησιμοποιώντας τις FFT ρουτίνες της DSPLIB, ακολουθώντας τα παρακάτω βήματα.

Πριν υλοποιήσετε τον φασματικό αναλυτή, γράψτε μία ρουτίνα σε γλώσσα C, που υπολογίζει τον FFT ενός αποθηκευμένου διανύσματος. Για την υλοποίηση του FFT θα πρέπει να ενημερωθείτε για τις παρακάτω συναρτήσεις της DSPLIB.   
– bitrev\_cplx (ή DSP\_ bitrev\_cplx)

– radix2 (ή DSP\_ radix2)

1. Η βιβλιοθήκη DSPLIB βρίσκεται στο αρχείο: C:\ti\c6000\dsplib\lib\dsp62x.lib το οποίο πρέπει να προσθέσετε στο project. Τα header αρχεία τα οποία πρέπει να προσθέσετε στον κώδικά σας βρίσκονται στο κατάλογο C:\ti\c6000\dsplib\include.

2. Διαβάστε το TMS320C62x DSP Library Programmer's Reference (SPRU402 ή SPRU402a) για να μάθετε πώς να χρησιμοποιείτε τις συναρτήσεις.

3. Χρησιμοποιώντας τις συναρτήσεις της βιβλιοθήκης, γράψτε κώδικα σε C ο οποίος θα υπολογίζει τον FFT χρησιμοποιώντας τον radix2 (ή DSP\_ radix2) αλγόριθμο όταν τα δεδομένα είναι αποθηκευμένα σε ένα διάνυσμα.

Βοηθητικές πληροφορίες: Ένα από τα δεδομένα εισόδου της συνάρτησης radix2 είναι ένα διάνυσμα που περιέχει τα εκθετικά, w, που χρησιμοποιεί ο FFT. Θα πρέπει να υπολογίσετε τις κατάλληλες τιμές του ημιτόνου και του συνημιτόνου και να τις αποθηκεύσετε σε Q-15 format. Για να χρησιμοποιήσετε τις συναρτήσεις ημιτόνου και συνημιτόνου θα πρέπει να συμπεριλάβετε στον κώδικα το αρχείο κεφαλής math.h. Ο υπολογισμός θα γίνει ως εξής:

for (i = 0; i < nx/2; i++)

{ w [i\*2] = 32767 \* -cos( i\*2\*pi/nx );

w [i\*2 + 1] = 32767 \* -sin( i\*2\*pi/nx );

}

4. Δημιουργείστε 64 δείγματα από μία συνάρτηση ημίτονου 10kHz που έχει υποστεί δειγματοληψία με ρυθμό 48kHz και αποθηκεύστε τα σε ένα διάνυσμα μήκους 64 σε Q-15 format. Υπολογίστε τον FFT του διανύσματος χρησιμοποιώντας τον κώδικά σας, και αποθηκεύστε το τετράγωνο του μέτρου του αποτελέσματος σε ένα διάνυσμα. Χρησιμοποιώντας το code composer studio δείτε το αποτέλεσμα. Βλέπετε μεγάλη τιμή στο σημείο που αντιστοιχεί στα 10kHz;

5. Αν το αποτέλεσμα του FFT δεν φαίνεται καλό, τότε υπάρχει μεγάλη πιθανότητα να έχει συμβεί λάθος λόγω υπερχείλισης. Για να αποφύγετε τα λάθη υπερχείλισης, θα πρέπει να κανονικοποιήσετε κατάλληλα το σήμα εισόδου. Προσπαθήστε πολλαπλασιάζοντας το σήμα εισόδου (με αριθμό < 1) με διάφορες τιμές κανονικοποίησης μέχρι να βρείτε μία κατάλληλη τιμή.

6. Κάνετε profiling στην συνάρτηση FFT και δείτε πόσο χρόνο παίρνει ο υπολογισμός του FFT μήκους 64. Ποιος είναι ο μέγιστος ρυθμός δειγματοληψίας που μπορεί να «αντέξει» ο FFT αν τον χρησιμοποιούμε συνεχώς για τα δείγματα εισόδου αποθηκεύοντάς τα σε μπλοκ των 64 σημείων.

Τώρα θα υλοποιήσετε έναν αναλυτή φάσματος. Θα χρειαστείτε τέσσερις buffers μήκους 64. Οι συναρτήσεις για κάθε buffer και η ροή του προγράμματος είναι όπως ακολουθεί (Εικόνα 2). Θα πρέπει να προσέξετε πολύ τον χρόνο που παίρνει κάθε μεταφορά δεδομένων.

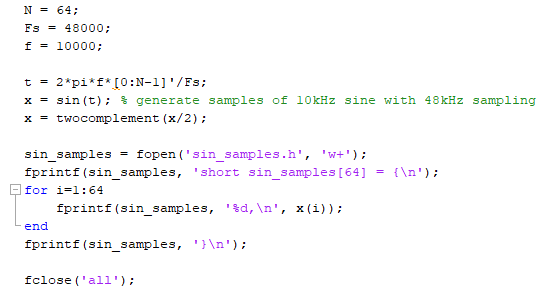
1. Η συνάρτηση εξυπηρέτησης αίτησης διακοπής εισόδου σήματος αποθηκεύει τα δείγματα εισόδου σε ένα buffer εισόδου.

2. Όταν έχουν μαζευτεί 64 δείγματα, το περιεχόμενο του buffer αντιγράφεται στον FFT buffer εισόδου. Τότε καλείτε την ρουτίνα FFT και υπολογίζετε τον FFT των δειγμάτων του FFT input buffer και αποθηκεύετε το αποτέλεσμα στον FFT output buffer.

3. Η συνάρτηση εξυπηρέτησης αίτησης διακοπής εξόδου βγάζει το αποτέλεσμα (αφού υπολογίσει το μέτρο) του codec output buffer, στον codec. Όταν όλες οι τιμές (64) βγουν στην έξοδο, αντιγράψτε το αποτέλεσμα του FFT output buffer στον codec output buffer.

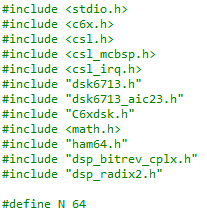
Αρχικά, αφού μελετήθηκε η λειτουργία των συναρτήσεων που δίνονται, μέσω του TMS320C62x DSP Library, και ρυθμίστηκε κατάλληλα το project, δημιουργήθηκε, μέ τo matlab, header αρχείο ημίτονου 10kHz που έχει υποστεί δειγματοληψία με ρυθμό 48kHz και αποθηκεύστε τα σε ένα διάνυσμα μήκους 64 σε Q-15 format (Lab8\_3.m).

Δημιουργείται αρχικά το σήμα του ημιτόνου και μετά με την twocomplement και με μία διαίρεση με το 2 όλων των συντελεστών, για να ανήκουν στο διάστημα [1,1), υπολογίζονται οι συντελεστές σε Q-15 format και τυπώνονται στο αρχείο sin\_samples.h .

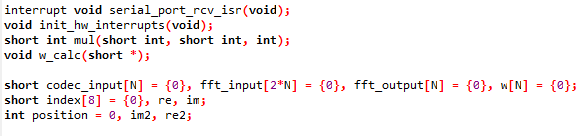


Με την βοηθητική συνάρτηση w, υπολογίστηκε ο fft, πολλαπλασιάζοντας αυτή τη φορά τους συντελεστές με το 2, για το Q-15 format. Ο αλγόριθμος αυτός αποτέλεσε την βάση για τον αναλυτή φάσματος που ακολουθεί (main\_lab8.c).

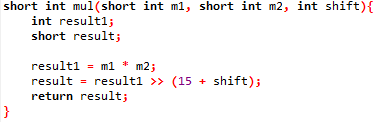
Στην αρχή, γίνονται οι απαραίτητες συμπεριλήψεις αρχείων και ορίζεται η μεταβλητή: Ν ίση με 64.



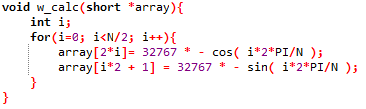
Έπειτα, ορίζονται οι συναρτήσεις και οι μεταβλητές που χρησιμοποιούνται στην πορεία του υπολογισμού.



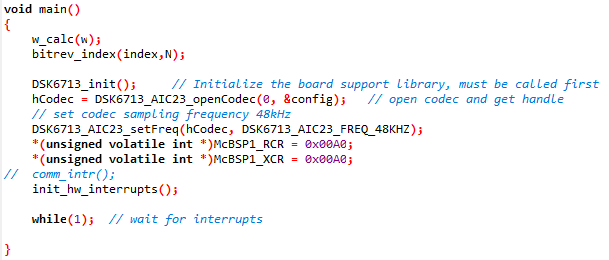
Η συνάρτηση mul() επιτελεί τον πολλαπλασιασμό των δύο πρώτων της ορισμάτων και μετατρέπει το αποτέλεσμα σε Q-15 format. Επίσης, μπορεί να κάνει και κάποια επιπλέον ολίσθηση μέσω του τρίτου ορίσματος shift που προστίθεται στο shift 15 bits που γίνεται ούτως ή άλλως.



Η w\_calc() υπολογίζει τον πίνακα w όπως ακριβώς υποδεικνύει η εκφώνηση της άσκησης.

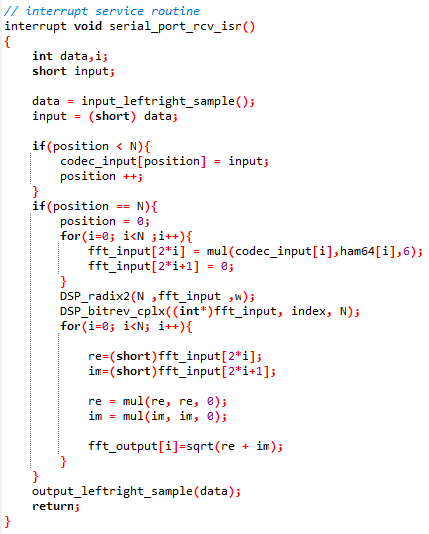


Στην main() συνάρτηση γίνονται οι ίδιες αρχικοποιήσεις με τις προηγούμενες ασκήσεις και επίσης, υπολογίζεται ο βοηθητικός πίνακας w, και ο πίνακας index, μέσω της bitrev\_index(), ο οποίος περιέχει τα indexes για την dsp\_bitrev\_cplx() που καλείται στην συνέχεια και υλοποιεί την απαραίτητη αντιστροφή των bits για να αναπαρίστανται σωστά τα πραγματικά και φανταστικά μέρη στο τέλος. Στη συνέχεια, η main() τρέχει έναν ατέρμονα βρόχο για να εξυπηρετεί το interrupt service routine.



Στο interrupt γίνεται ο υπολογισμός του fft. Κάθε φορά, λαμβάνεται η είσοδος μέσω της input\_leftright\_sample() και αποθηκεύεται στην μεταβλητή data. Έπειτα, γίνεται cast σε short int και αποθηκεύεται στην μεταβλητή input. Η μεταβλητή position δείχνει τη θέση, κάθε φορά, του πίνακα εισόδου codec\_input[ ]. Όσο είναι μικρότερη του 64, γράφεται η είσοδος στον πίνακα εισόδου, και αυξάνεται κατά ένα στην συνέχεια. Όταν γίνει 64, η position αρχικοποιείται ξανά ίση με 0, για τα επόμενα interrupts, και αρχίζει ο υπολογισμός του fft, με γεμάτο πλέον πίνακα εισόδου.

Με μία επανάληψη, πολλαπλασιάζονται τα στοιχεία του πίνακα εισόδου με τον πίνακα συντελεστών ham64. Τα αποτελέσματα αποθηκεύονται στις ζυγές θέσεις του πίνακα fft\_input, που θα αναπαραστήσουν το πραγματικό μέρος του τελικού μετασχηματισμού, ενώ στις περιττές θέσεις αποθηκεύεται το 0. Αυτές οι θέσεις θα αναπαραστήσουν το φανταστικό μέρος του μετασχηματισμού. Επίσης, θέτοντας το τρίτο όρισμα του πολλαπλασιασμού ίσο με 6, γίνεται και μία διαίρεση με το 64 στο αποτέλεσμα για να αποφευχθούν οι υπερχειλίσεις. Σημειώνεται ότι παρατηρήθηκε πως ακόμα και με το 32 ή το 16 ή το 8 να γινόταν η διαίρεση, πάλι προέκυπτε σωστό αποτέλεσμα. Στη συνέχεια, καλείται η dsp\_radix2() που υπολογίζει τον fft με δοσμένο το μέγεθος, αλλά επιστρέφει το αποτέλεσμα σε bit-reversed σειρά. Την επαναφορά των bit στη σωστή θέση κάνει η dsp\_bitrev\_cplx() με τη βοήθεια του πίνακα index[ ] που ορίστηκε στην αρχή. Τέλος, με μία επανάληψη, λαμβάνεται κάθε φορά δύο αριθμοί από τον fft\_input[ ], πραγματικό και φανταστικό μέρος, πολλαπλασιάζονται με τον εαυτό τους για να υπολογιστεί το τετράγωνο τους και αποθηκεύεται η ρίζα του αθροίσματος των τετραγώνων (μέτρο μετασχηματισμού) στον fft\_output[ ].



Θέτοντας είσοδο ένα ημιτονοειδές σήμα 1ΚHz, παρατηρείται στα αρχικά στάδια του προγράμματος το παρακάτω αποτέλεσμα:

Εικόνα που περιέχει στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Οι κορυφές παρατηρούνται στο δείγμα +1 και στο +63, που είναι αναμενόμενο. Το πρόγραμμα μπορεί να σταματήσει εδώ αφού έχει υπολογιστεί ο πίνακας εξόδου. Αν συνεχιστεί, παρόλα αυτά, στη συνέχεια το γράφημα του πίνακα:

Εικόνα που περιέχει στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Εδώ παρατηρούνται στα +1, +63, +65, +126 περίπου. Τέλος, λαμβάνει την μορφή:

Εικόνα που περιέχει στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Παρατηρείται συμμετρία στις κορυφές, η οποία καθορίζεται από τον αριθμό δειγμάτων και την συχνότητα εισόδου. Φυσικά και από τη συχνότητα δειγματοληψίας.

**Βιβλιογραφία:**

* **TMS320C67x/C67x+ DSP CPU and Instruction Set Reference Guide**
* **ΕΚΦΩΝΗΣΗ ΑΣΚΗΣΗΣ LAB-8 – ECLASS**
* **TMS320C6713 DATASHEET**
* **TMS320C62x DSP Library**