

ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ

Εργαστήριο 7

Ομάδα 09 – Group 1

ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ
ΔΑΣΟΥΛΑΣ	ΙΩΑΝΝΗΣ	1053711
ΔΟΥΡΔΟΥΝΑΣ	ΑΡΙΣΤΕΙΔΗΣ ΑΝΑΓΥΡΟΣ	1047398

Υλοποίηση DTMF αποκωδικοποιητή

Στόχος:

Σε αυτό το εργαστήριο θα υλοποιηθεί ένα Dual-Tone Multi-Frequency(DTMF) αποκωδικοποιητή στην πλατφόρμα TMS320C6713 χρησιμοποιώντας Fixed-point αριθμητική.

Ασκήσεις:

Άσκηση 7.1

Να γραφεί μία συνάρτηση που να υλοποιεί τον αλγόριθμο Goertzel σε Matlab. Να ελεγχθεί τη σωστή λειτουργία του αλγορίθμου παράγοντας κατάλληλους DTMF τόνους. Για τη δημιουργία τους να χρησιμοποιηθούν οι συναρτήσεις ημίτονου της Matlab.

Αναλυτικά, η υλοποίηση του DTMF αποκωδικοποιητή με τη χρήση του αλγορίθμου Goertzel ακολουθεί τα παρακάτω βήματα:

•Για όλους τους συνδυασμούς συχνοτήτων γραμμής-στήλης της Εικόνας 1 συντίθεται οι (16 διαφορετικούς) ημιτονοειδείς ήχοι των πλήκτρων.

•Για κάθε ένα από τα παραγόμενα σήματα εισόδου (τόνους) και για κάθε μία από τις 8 διαφορετικές συχνότητες του DTMF εκτελείται ο αλγόριθμος Goertzel:

- Για $N=205$ επαναλήψεις (άρα και 205 δείγματα του σήματος εισόδου $x(n)$) εκτελείται ο αναδρομικός τύπος (10) και υπολογίζονται τα Q_n .
- Χρησιμοποιώντας τις δύο τελευταίες τιμές του Q_n υπολογίζεται το τετράγωνο του μέτρου της φάσης $(|y_k(N)|^2)$ από την εξίσωση (14).

•Οι 2 συχνότητες (από τις 8) για τις οποίες ο αλγόριθμος Goertzel επιστρέφει τις μεγαλύτερες τιμές του τετραγώνου του μέτρου της φάσης, δηλώνουν την “ταυτότητα” του τόνου που ανιχνεύεται κάθε φορά.

		"High Group" frequencies [Hz]			
		1209	1336	1477	1633
"Low Group" frequencies [Hz]	697	1	2	3	A
	770	4	5	6	B
	852	7	8	9	C
	941	*	0	#	D

Εικόνα 1. Αντιστοιχία συμβόλων και τόνων

Ο κώδικας που αναπτύχθηκε ,σε Matlab, για το ερώτημα 7.1 είναι ο ακόλουθος:

```

2 - f_lo = [697 770 852 941];
3 - f_hi = [1209 1336 1477 1633];
4
5 - spectrum_lo=[0 0 0 0];
6 - spectrum_hi=[0 0 0 0];
7
8 - coef_lo = [1.703275 1.635585 1.562297 1.482867];
9 - coef_hi = [1.163138 1.008835 0.790074 0.559454];
10
11 - buttons = ['1','2','3','A' ; '4','5','6','B' ; '7','8','9','C' ; '*','0','#','D'];
12
13 - for i=1:4 ...
31
32
33 - function f = signal(x) ...
38
39 - function Y = Goertzel(x,coef)
40 -     Qn_1 = 0;
41 -     Qn_2 = 0;
42 -     N = 205;
43 -     for l = 1:N
44 -         Q_n = x(l)+ coef *Qn_1 - Qn_2;
45 -         Qn_2 = Qn_1;
46 -         Qn_1 = Q_n;
47 -         t = coef*Qn_1;
48 -     end
49 -     Y = Q_n^2 + Qn_1^2 - coef * Q_n * Qn_1;
50 - end
51

```

Στο πρώτο κομμάτι του κώδικα αποθηκεύονται σε στους πίνακες `f_lo` και `f_hi` οι τιμές των χαμηλών και υψηλών συχνοτήτων αντίστοιχα έτσι ώστε να μπορεί να γίνει η αντιστοίχιση με τα πλήκτρα. Στη συνέχεια αποθηκεύονται στους πίνακες `coef_lo` και `coef_hi` οι συντελεστές των χαμηλών και των ψηλών συχνοτήτων ενώ στον πίνακα `buttons` που είναι 4X4, αποθηκεύονται οι τιμές των κουμπιών. Τέλος αρχικοποιούνται οι πίνακες `spectrum_lo` και `spectrum_hi` οι στους οποίους αποθηκεύονται οι τιμές του φάσματος των συχνοτήτων του κάθε δείγματος εισόδου.

Στη συνέχεια δημιουργήθηκαν δύο συναρτήσεις :

```
f = signal(x)
Y = Goertzel(x,coef)
```

Η πρώτη συνάρτηση χρησιμοποιείται για να δημιουργήσει το σήμα εισόδου. Πιο συγκεκριμένα δημιουργεί ένα σήμα ημιτόνου για 205 δείγματα με συχνότητα δειγματοληψίας ίση με 8 kHz. Ως αποτέλεσμα , λαμβάνονται όλοι οι δυνατοί συνδυασμοί των συχνοτήτων που αντιστοιχούν στα κουμπία.

Η δεύτερη συνάρτηση που χρησιμοποιήθηκε υλοποιεί τον αλγόριθμο του Goertzel όπως περιγράφεται στην εκφώνηση του εργαστηρίου. Δέχεται ως ορίσματα καθέναν από 4 συντελεστές και το σήμα εισόδου `x`. Συγκεκριμένα, υπολογίζεται ο τύπος $Q(n)=x(n)+2\cos(2\pi kN)Q(n-1)-Q(n-2)$. Τα $Q(n-1)$, $Q(n-2)$ αποθηκεύονται στις μεταβλητές `Qn_1` , `Qn_2` και στο `Q_n` υπολογίζεται ο παραπάνω τύπος για 205 επαναλήψεις. Έπειτα, το `Qn_2` παίρνει την τιμή της επόμενης κατάστασης (`Qn_1`) και αντίστοιχα το `Qn_1` παίρνει την τιμή του `Q_n`. Με το τέλος των 205 επαναλήψεων υπολογίζεται του μέτρο του σήματος με βάση τον τύπο: $|y(k)|^2=|X(k)|^2=Q^2(N)+Q^2(N-1)-2\cos(2\pi kN)Q(N)Q(N-1)$.

```
33 function f = signal(x)
34 -     Fs = 8000;
35 -     N = 205;
36 -     f = sin(2*pi*x*(1:N+1)/Fs);
37 - end
38
39 function Y = Goertzel(x,coef)
40 -     Qn_1 = 0;
41 -     Qn_2 = 0;
42 -     N = 205;
43 -     for l = 1:N
44 -         Q_n = x(l)+ coef *Qn_1 - Qn_2;
45 -         Qn_2 = Qn_1;
46 -         Qn_1 = Q_n;
47 -     end
48 -     Y = Q_n^2 + Qn_1^2 - coef * Q_n * Qn_1;
49 - end
50 -
```

Στην κύρια επανάληψη του προγράμματος υπάρχουν 2 for loops από 1-4 έτσι ώστε το πρόγραμμα να εκτελεστεί για κάθε μία από τις low και high συχνότητες

αντίστοιχα. Στη συνέχεια δημιουργείται το σήμα που αποτελείται από το άθροισμα 2 ημιτόνων, τα οποία δημιουργούνται μέσω της `signal`. Έπειτα μέσα σε μία `for` 4 επαναλήψεων καλείται η συνάρτηση του αλγορίθμου για κάθε έναν από τους συντελεστές. Τα φάσματα της υψηλής και της χαμηλής συχνότητας αποθηκεύονται στους πίνακες `spectrum_lo` και `spectrum_hi` αντίστοιχα που καλούν την συνάρτηση Goertzel με πρώτο όρισμα την είσοδο και δεύτερο όρισμα τους συντελεστές των χαμηλών και των υψηλών συχνοτήτων. Στη συνέχεια προκειμένου να γίνει η ανίχνευση του σωστού κουμπιού που πατήθηκε, μέσω της εντολής `max`, εντοπίζεται η μέγιστη συχνότητα που υπάρχει στους πίνακες `spectrum_lo` και `spectrum_hi`. Τέλος ανάλογα με τις μέγιστες τιμές που βρίσκονται στον πίνακα παίρνουν τιμές τα `low` και `high` έτσι ώστε να τυπωθεί το αντίστοιχο κουμπί που έχει συντεταγμένες (`low,high`) . Έτσι, τυπώνονται όλα τα κουμπιά του πληκτρολογίου με την σειρά. Η έξοδος του προγράμματος είναι η ακόλουθη:

```
>> Lab7_1
    1    2    3    A
    4    5    6    B
    7    8    9    C
    *    0    #    D
fx >>
```

Άσκηση 7.2 (υλοποίηση DTMF αποκωδικοποιητή στον TMS320C6713)

Χρησιμοποιώντας ως βάση το πρόγραμμα που υλοποιήσατε για την εργαστηριακή άσκηση 1 ή 2 (δηλαδή είτε με `polling`, είτε με `interrupts`), υλοποιήσετε ένα πρόγραμμα που θα αναγνωρίζει DTMF τόνους χρησιμοποιώντας `fixed point` αριθμητική. Προτείνεται να χρησιμοποιήσετε τον παραπάνω πίνακα για τις τιμές του συντελεστή, οπότε θα πρέπει να κάνετε υποδειγματοληψία στα δείγματα που παίρνετε, αφού δουλεύετε στα $48kHz$. Καθώς οι τιμές του πίνακα δίνονται για συχνότητα $f_s = 8kHz$, θα πρέπει να εφαρμόσετε κατάλληλη υποδειγματοληψία κατά $(1/6)$. Χρησιμοποιείστε ένα κινητό τηλέφωνο ή κατάλληλο λογισμικό για την παραγωγή των DTMF τόνων.

Στο πρόγραμμα που αναπτύχθηκε, αρχικά φορτώθηκαν τα απαραίτητα αρχεία και έγιναν οι αρχικοποιήσεις μεταβλητών και συναρτήσεων που χρησιμοποιούνται στη συνέχεια.

```
interrupt void serial_port_rcv_isr(void);
short int mul(short int , short int, int );
void init_hw_interrupts(void);
short int Goertzel(short int* , short int, int );
int buffer_max( short int *);

int N = 205, current = 0;
short int buffer_in[205] = {0};
int f_lo[4] = {697, 770, 852, 941}, f_hi[4] = {1209, 1336, 1477, 1633};
short coef_lo[4] = {0x6D02, 0x68AD, 0x63FC, 0x5EE7};
short coef_hi[4] = {0x4A70, 0x4090, 0x6521, 0x479C};
const char buttons[4][4] = {{ '1', '2', '3', 'A'}, {'4', '5', '6', 'B'},
                             {'7', '8', '9', 'C'}, {'*', '0', '#', 'D'}};
short int temp0 ,temp1 ,temp2 ,temp3, y;
```

Στην main συνάρτηση γίνεται μόνο η αλλαγή της συχνότητας δειγματοληψίας, από 48KHz σε 8KHz, αλλάζοντας το δεύτερο όρισμα της DSK6713_AIC23_setFreq().

```
void main()
{
    DSK6713_init();      // Initialize the board support library, must be called first
    hCodec = DSK6713_AIC23_openCodec(0, &config); // open codec and get handle
    // set codec sampling frequency 8kHz
    DSK6713_AIC23_setFreq(hCodec, DSK6713_AIC23_FREQ_8KHZ);
    *(unsigned volatile int *)McBSP1_RCR = 0x00A0;
    *(unsigned volatile int *)McBSP1_XCR = 0x00A0;
    // comm_intr();
    init_hw_interrupts();
    while(1); // wait for interrupts
}
```

Χρησιμοποιούνται, επίσης, τρεις βοηθητικές συναρτήσεις. Η πρώτη, η mul(), επιτελεί τον πολλαπλασιασμό συντελεστών σε Q-15 format χρησιμοποιώντας ένα όρισμα d στο οποίο δίνεται πάντα η τιμή 0 ή 1. Η τιμή 0 δίνεται όταν δεν χρειάζεται και ένας επιπλέον πολλαπλασιασμός επί 2, δηλαδή για συντελεστές που δεν έχουν διαιρεθεί με το 2 στον πίνακα. Αντίθετα, η τιμή 1 δίνεται όταν στον πολλαπλασιασμό συμμετέχει συντελεστής που αρχικά έχει διαιρεθεί με το 2, ώστε να γίνει ένα shift αριστερά, δηλαδή ένας πολλαπλασιασμός με το 2.

```
short int mul(short int m1, short int m2, int d){
    short int result;

    result = (short) ( (m1 * m2) >> (15 - d));

    return result;
}
```

Η συνάρτηση `buffer_max` δέχεται ως όρισμα έναν πίνακα `buffer` και με μία επανάληψη, επιστρέφει την θέση της μέγιστης τιμής του πίνακα.

```
int buffer_max( short int *buffer){
    int i, index = 0;
    short int max = 0;

    for(i=0; i<4; i++){
        if(buffer[i] > max){
            max = buffer[i];
            index = i;
        }
    }
    return index;
}
```

Η συνάρτηση `Goertzel()` δέχεται ως όρισμα τον πίνακα εισόδου `buffer`, τον πίνακα συντελεστών του `e-class`: `coef` και τον αριθμό `d` που είναι απαραίτητος για την `mul()` συνάρτηση. Με έναν βρόχο 205 επαναλήψεων επιτελείται ο αλγόριθμος και αποθηκεύονται κάθε φορά οι προηγούμενες τιμές, εκτός από την τελευταία επανάληψη, ώστε να μην υπάρχουν στο τέλος δύο ίδιες τιμές. Στο τέλος προκύπτει το τετράγωνο του μέτρου, σύμφωνα με τον τύπο του `eclass`.

```
short int Goertzel(short int *buffer, short coef, int d){
    int i;
    short int q[3] = {0};

    for(i=0; i<N; i++){
        q[0] = buffer[i] + mul(q[1], coef, d) - q[2];
        q[2] = q[1];
        if(i<N-1) q[1] = q[0];
    }

    temp0 = mul(q[0], q[0], 0);
    temp1 = mul(q[1], q[1], 0);
    temp2 = mul(coef, q[0], d);
    temp3 = mul(temp2, q[1], 0);

    y = temp0 + temp1 - temp3;
    return y;
}
```

Στο interrupt service routine συνδυάζονται τα παραπάνω για να παραχθεί το αποτέλεσμα. Κάθε φορά που καλείται το interrupt, αποθηκεύεται η είσοδος `data` και γίνεται cast στην `short int` μεταβλητή `input`. Έπειτα, η `input` διαιρείται με το 4 με μία `shift` εντολή, για την αποφυγή υπερχειλίσσεων, και αν τελικά η `input` είναι μεγαλύτερη μιας στάθμης, που στο πρόβλημα τέθηκε μετά από δοκιμές ίση με 5, η τιμή της αποθηκεύεται στον πίνακα `buffer_in` όσο αυτός έχει ακόμα κενές θέσεις. Η στάθμη τίθεται για να μην μπαίνει θόρυβος στον πίνακα τιμών. Όταν ο `buffer_in` γεμίσει, υπολογίζεται το φάσμα καλώντας την συνάρτηση `Goertzel()` όπως ακριβώς περιεγράφηκε και στο πρόγραμμα του Matlab. Έπειτα, με την `buffer_max` βρίσκεται το `index` της μέγιστης τιμής του πίνακα του φάσματος, για χαμηλές και υψηλές συχνότητες αντίστοιχα. Γνωρίζοντας το `max index`, μπορεί να τυπωθεί έπειτα το

σωστό κουμπί. Τέλος, ο μετρητής στοιχείων του πίνακα μηδενίζεται για να επαληφθεί η διαδικασία.

```
// interrupt service routine
interrupt void serial_port_rcv_isr()
{
    int data, i, d, index_lo, index_hi;
    short int input, spectrum_lo[4] = {0}, spectrum_hi[4] = {0};

    data = input_leftright_sample();
    input = (short) data;
    input = input >> 2;
    if( input > 5 || input < -5){

        if(current < N){
            buffer_in[current] = input;
            current ++;

        }if(current == N){

            for(i=0; i<4; i++){
                if(i>1) d = 0;
                else d = 1;

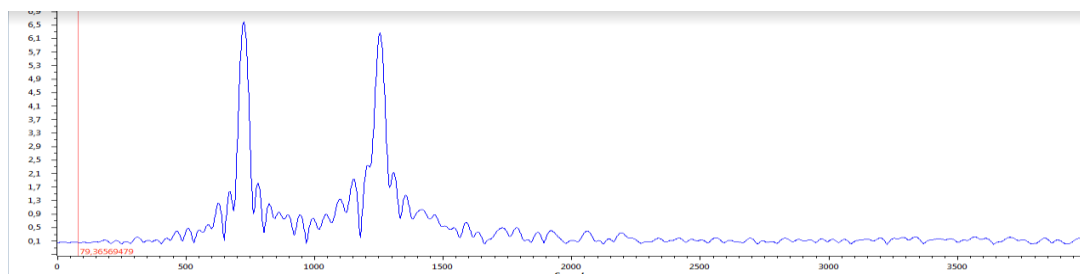
                spectrum_lo[i] = Goertzel(buffer_in, coef_lo[i], 1);
                spectrum_hi[i] = Goertzel(buffer_in, coef_hi[i], d);
            }

            index_lo = buffer_max(spectrum_lo);
            index_hi = buffer_max(spectrum_hi);

            printf(" Button: %c Low frequency: %d, High frequency: %d\n",
                buttons[index_lo][index_hi], f_lo[index_lo], f_hi[index_hi]);
            current = 0;
        }
    }
}
```

Παρόλο που όπως φαίνεται στην φωτογραφία, η είσοδος αποθηκευόταν σωστά (στην φωτογραφία για πάτημα του κουμπιού 1 αποτυπώνεται ο fft του buffer_in) και παρόλο που ο αλγόριθμος είναι σωστός, μιας και η ορθότητά του επιβεβαιώθηκε στο πρόγραμμα του Matlab, το πρόγραμμα δεν είχε πάντα το σωστό αποτέλεσμα, πιθανότατα λόγω υπερχειλίσεων. Τις περισσότερες φορές, πάντως, προέκυπτε το σωστό κουμπί.

FFT εισόδου για συνεχές πάτημα του κουμπιού 1:



Ενδεικτικά αποτελέσματα για συνεχές πάτημα του κουμπιού 1:

```
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 2 Low frequency: 697, High frequency: 1336
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 4 Low frequency: 770, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 4 Low frequency: 770, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: * Low frequency: 941, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 2 Low frequency: 697, High frequency: 1336
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: * Low frequency: 941, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 7 Low frequency: 852, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
Button: 1 Low frequency: 697, High frequency: 1209
```

Βιβλιογραφία:

- **TMS320C67x/C67x+ DSP CPU and Instruction Set Reference Guide**
- **ΕΚΦΩΝΗΣΗ ΑΣΚΗΣΗΣ LAB-7 – ECLASS**
- **TMS320C6713 DATASHEET**
- **AIC23 Data Manual**

