



Πανεπιστήμιο Πειραιώς - Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών “Πληροφορική”

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Σχεδιασμός και υλοποίηση ενσωματωμένου συστήματος ψηφιακού Delay-Echo για ακουστικό σήμα
Όνοματεπώνυμο Φοιτητή	Μίσσης Θωμάς
Πατρώνυμο	Δημότριος
Αριθμός Μητρώου	ΜΠΠΑ 08045
Επιβλέπων	Άγγελος Πικράκης, Δέκτωρ

РАНЕКЪТЪМО ТЕРАА

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΠΑ

Τριμελής Εξεταστική Επιτροπή

Α. Πιπιάκης Δέκτορας	Μ. Ψιπιάκης Κπ. Καθηγητής	Δ. Γκιζόπουλος Αν. Καθηγητής

РАНЕКЪТЪМО ТЕРАА

Περιεχόμενα

I	Θεωρητικό Μέρος	1
1	Εισαγωγή	2
1.1	Σύντομη περιγραφή του προβλήματος	2
2	Γραμμές Καθυστέρησης (Delay Lines)	5
2.1	Εισαγωγή	5
2.2	Ψηφιακά Φίλτρα	5
2.2.1	Feedforward Comb Filter	5
2.2.2	Feedback Comb Filter	8
2.2.3	All-Pass Filter	11
3	Αντήησηση (Reverb)	12
3.1	Φυσική Αντήησηση	12
3.2	Ψηφιακά συστήματα Τεχνητής Αντήησησης	13
II	Σχεδίαση ενσωματωμένου συστήματος ψηφιακού Delay-Echo	16
4	Μεθοδολογία ανάπτυξης του συστήματος	17
4.1	Φάσεις Ανάπτυξης	17
5	Περιγραφή του συστήματος Digital Delay	20
5.1	Διαγραμματική μέθοδος ανάλυσης συστημάτων IDEF0	20
5.2	Λειτουργική περιγραφή του ψηφιακού συστήματος Delay-Echo	20
5.3	Ανάλυση του ψηφιακού υποσυστήματος	21
5.3.1	A-D μετατροπέας (MCP3202)	21
5.3.2	D-A μετατροπέας (MCP1821)	28
5.3.3	Ο μικροελεγκτής Propeller	30
5.3.4	Υπόσυστημα ελέγχου των παραμέτρων Feedback gain, Delay time.	35
5.3.5	Δοκιμικό ανάπτυξης	37
5.4	Ανάλυση και Αναλογιστικό υποσυστήματος	42
6	Συμπεράσματα-Περαιτέρω	46
	Βιβλιογραφία	48

Κατάλογος σχημάτων

1.1	Μοντέλο ψηφιακού ηχητικού επέ με σύστημα ελέγχου	3
1.2	Το κλασσικό Buss Digital Delay	3
2.1	Διάγραμμα ροής σήματος για ένα απλό Feedforward φίλτρο.	6

2.2	Γραφική αναπαράσταση των μιγαδικών ριζών ενός Feedforward Comb φίλτρου.	7
2.3	Απόκριση συχνότητας ενός Feedforward Comb φίλτρου.	7
2.4	Διάγραμμα ροής σήματος για ένα Feedback φίλτρο.	8
2.5	Γραφική αναπαράσταση των μιγαδικών πόλων ενός Feedback Comb φίλτρου.	9
2.6	Απόκριση συχνότητας ενός Feedback Comb φίλτρου με feedback gain $g > 0$.	9
2.7	Κρουστική Απόκριση ενός Feedback Comb φίλτρου.	10
2.8	Διάγραμμα ροής σήματος για ένα All-Pass φίλτρο.	11
2.9	Κρουστική απόκριση για ένα All Pass φίλτρο πρώτης τάξης.	11
3.1	Φυσική Αντήρηση σε έναν κλειστό χώρο	12
3.2	Κρουστική απόκριση ενός κλειστού χώρου	13
3.3	Δομή ενός δημοφιλούς συστήματος τεχνητής αντήρησης, από τον James Moorer.	14
3.4	Προσθήκη ενός βαθυτερατού φίλτρου στο Feedback Loop ενός Comb φίλτρου	15
5.1	Activity box: Το δομικό στοιχείο της μεθόδου IDEF0.	20
5.6	MCP3202 pinout	21
5.2	Αποψη του υψηλότερου επιπέδου του συστήματος: Κόμβος A0	22
5.3	Αποσύνθεση κόμβου A0	23
5.4	Αποσύνθεση του Αναλογικού υποσυστήματος (Κόμβος A1)	24
5.5	Αποσύνθεση του Ψηφιακού υποσυστήματος (Κόμβος A2)	25
5.7	Λειτουργικό διάγραμμα του MCP3202	26
5.8	Configuration bits για το MCP3202	27
5.9	Επικοινωνία με το MCP3202.	27
5.10	MCP1821 pinout	28
5.11	Λειτουργικό διάγραμμα του MCP1821	29
5.12	Επικοινωνία με το MCP1821	29
5.13	Καταχωρητές εντολής εγγραφής για το MCP1821	30
5.14	Block διάγραμμα για το Propeller chip	32
5.15	Οι καταχωρητές ειδικών λειτουργιών για ένα cog	33
5.16	Παράδειγμα συνδεσμολογίας του Propeller chip με εσωτερική μνήμη EEPROM και ένα 2C	34
5.17	Μέτρηση του χρόνου εκφόρτισης πυκνωτού μέσω μεταβλητής αντιστάσεως	35
5.18	Βελτιωμένο κύκλωμα σύνδεσης ποτενσιομέτρου με το Propeller chip	36
5.19	Ηλεκτρονικό κύκλωμα του ψηφιακού υποσυστήματος	38
5.20	Δομή του προγράμματος σε σχέση με το χρησιμοποιούμενο hardware	39
5.21	Κύκλωμα Sallen-Key	43
5.22	Ηλεκτρονικό κύκλωμα του αναλογικού υποσυστήματος	45

Κατάλογος πινάκων

4.1	Αρχικές απαιτήσεις-προδιαγραφές του συστήματος	19
5.1	Μερικά τεχνικά χαρακτηριστικά του MCP3202	26
5.2	Μερικά τεχνικά χαρακτηριστικά του MCP1821	30
5.3	Μερικά τεχνικά χαρακτηριστικά του μικροελεγκτή Propeller	34
5.4	Υπολογισμός συχνότητας δειγματοληψίας f_s	42
5.5	Τεχνικά χαρακτηριστικά για τα δύο βαθυτερατά φίλτρα (ADC, DAC)	43
6.1	Τελικές προδιαγραφές του συστήματος	47

Κατάλογος Ψευδοκώδικα

1	Διάβασμα των ποτενσιομέτρων από το Cog1 (Task: ReadPots)	40
2	Ο εκτελούμενος αλγόριθμος Echo από το Cog0	41
3	Αλγόριθμος πολλαπλασιασμού μεταξύ ενός προσημασμένου ακεραίου X (16 bits) και ενός μη προσημασμένου ακεραίου Y (9 bits)	42

Ευχαριστίες

Από την θέση αυτή, θα ήθελα να ευχαριστήσω τον Λέκτορα του Πανεπιστημίου Πειραιά, κ. Άγγελο Πικράκη, τόσο για την εμπιστοσύνη που μου έδειξε, όσο και για την επιστημονική βοήθεια που μου προσέφερε κατά την εκπόνηση αυτής της εργασίας. Επίσης θα ήθελα να ευχαριστήσω την εταιρία Microwarp Technology Inc. για την δωρεάν και έγκαιρη προμήθεια βασικού εξοπλισμού, γεγονός που επιτάχυνε σε μεγάλο βαθμό την διαδικασία της υλοποίησης του πρωτοτύπου. Τέλος ευχαριστώ θερμά τον συναδέλφο Ειμανουήλ Μανωλά για την πολύτιμη βοήθεια του ως προς το περιβάλλον ΕΠΧ, μέσω του οποίου έγινε η συγγραφή της παρούσας διατριβής.

Περίληψη

Σκοπός της παρούσας διατριβής είναι ο σχεδιασμός και η υλοποίηση ενός ενσωματωμένου συστήματος (Embedded System), που θα επεξεργάζεται σε πραγματικό χρόνο ένα ακουστικό σήμα προερχόμενο από ένα ηλεκτρικό μουσικό όργανο και θα δημιουργεί ένα εφέ καθυστέρησης (delay/echo effect). Τέτοια συστήματα χρησιμοποιούνται από μουσικούς και ηχολήπτες κατά τις διαδικασίες της ηχογράφησης και της μίξης, προκειμένου να δημιουργηθεί ένας πιο ρεαλιστικός ήχος, όπως επίσης και για την επίτευξη διαφόρων ηχητικών εφέ.

Το πρωτότυπο που υλοποιήσαμε αποτελείται από ένα ψηφιακό υποσύστημα και από ένα αναλογικό. Το ψηφιακό υποσύστημα συνίσταται από έναν προγραμματιζόμενο μικροελεγκτή και από δύο σειριακούς μετατροπείς σημάτων: έναν μετατροπέα αναλογικού σήματος σε ψηφιακό (ADC) και έναν μετατροπέα ψηφιακού σήματος σε αναλογικό (DAC).

Σχετικά με το αναλογικό υποσύστημα, αυτό περιλαμβάνει δύο όμοια βαθμιερατά φίλτρα. Το πρώτο χρησιμοποιείται ως antialiasing φίλτρο πριν το ADC, ενώ το δεύτερο φιλτράρει την έξοδο του DAC.

Το σύστημά μας βασίστηκε στον μικροελεγκτή *Propeller*. Το γεγονός ότι ο μικροελεγκτής αυτός δεν είναι ένας εξειδικευμένος ψηφιακός επεξεργαστής σήματος (DSP) αποτέλεσε μια επιπρόσθετη δυσκολία, ταυτόχρονα όμως και μια πρόκληση η οποία αντιμετωπίστηκε επιτυχώς.

Abstract

The aim of this thesis is the design and implementation of an embedded system, that is capable of processing in real time an acoustic signal, in order to produce a Delay-Echo effect. This kind of digital audio effect is useful, both to musicians and sound engineers. The goal of this system is to create a more realistic sound during recording sessions in a studio. Moreover, it can be used as an interesting musical effect.

Our prototype consists of a digital and an analog subsystem. The digital subsystem includes a microcontroller and two serial signal converters: an analog to digital (ADC) and a digital to analog one (DAC).

The analog subsystem is composed of two identical low pass filters. The first one is placed in front of the ADC as an antialiasing filter, while the second one filters the output of the DAC.

Our system is based on the "Propeller" microcontroller. The fact that this is not a Digital Signal Processor (DSP), consisted not only an additional constraint but also an algorithmic challenge which was successfully handled.

Μέρος Ι
Θεωρητικό Μέρος

Κεφάλαιο 1

Εισαγωγή

1.1 Σύντομη περιγραφή του προβλήματος

Στην ελοχή μας τα ηχητικά εφέ χρησιμοποιούνται ευρέως από όλους εκείνους που έχουν σχέση με την δημιουργία και επεξεργασία ηχητικών σημάτων, δηλαδή από απλούς μουσικούς που τα χρησιμοποιούν ως μέσο έκφρασης, μέχρι ηχολήπτες και παραγωγούς οι οποίοι τα χειρίζονται κατάλληλα προκειμένου να δώσουν κάποιο ιδιαίτερο χρώμα σε μια ηχογράφηση. Με τον όρο "Ψηφιακά Ήχητικά Εφέ" (Digital Audio Effects) εννοούμε συσκευές ή λογισμικό που επεξεργάζονται σήματα ήχου - σύμφωνα με κάποιες παραμέτρους ελέγχου που καθορίζονται από τον χρήστη - με αποτέλεσμα την δημιουργία ενός νέου σήματος εξόδου όπως φαίνεται στο σχήμα 1.1 [14]. Ο χρήστης έχει στην διάθεσή του την ακουστική και την οπτική αναπαράσταση των σημάτων εισόδου και εξόδου, προκειμένου να είναι σε θέση να αλληλεπιδρά με το σύστημα μεταβάλλοντας τις παραμέτρους ελέγχου.

Στόχος αυτής της εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός τέτοιου ψηφιακού ηχητικού εφέ, και συγκεκριμένα ενός συστήματος που θα προσδίδει στο σήμα εισόδου ένα εφέ καθυστέρησης (delay/echo). Η πηγή του σήματος εισόδου είναι ένα ηλεκτρικό μουσικό όργανο όπως για παράδειγμα μια ηλεκτρική κιθάρα ή ένα ηλεκτρικό μπάσο. Τέτοιες συσκευές κυκλοφορούν στο εμπόριο εδώ και πολλά χρόνια κυρίως με την μορφή "πεταλιών" (stomp boxes) από μεγάλες εταιρίες στον μουσικό χώρο, όπως η Roland, η Digitech, η Yamaha και άλλες. Στο σχήμα 1.2 βλέπουμε ένα δημοφιλές "πετάλι" για ηλεκτρική κιθάρα το οποίο πρωτοκυκλοφόρησε το 1983, όπου η τιμή του ήταν απαγορευτική ακόμα και για επαγγελματικά μουσικά. Λόγω του μεγάλου κόστους της μνήμης εκείνη την εποχή. Το μοντέλο έχει εξελιχθεί και αυτή τη στιγμή κυκλοφορεί η έκδοση DD6.

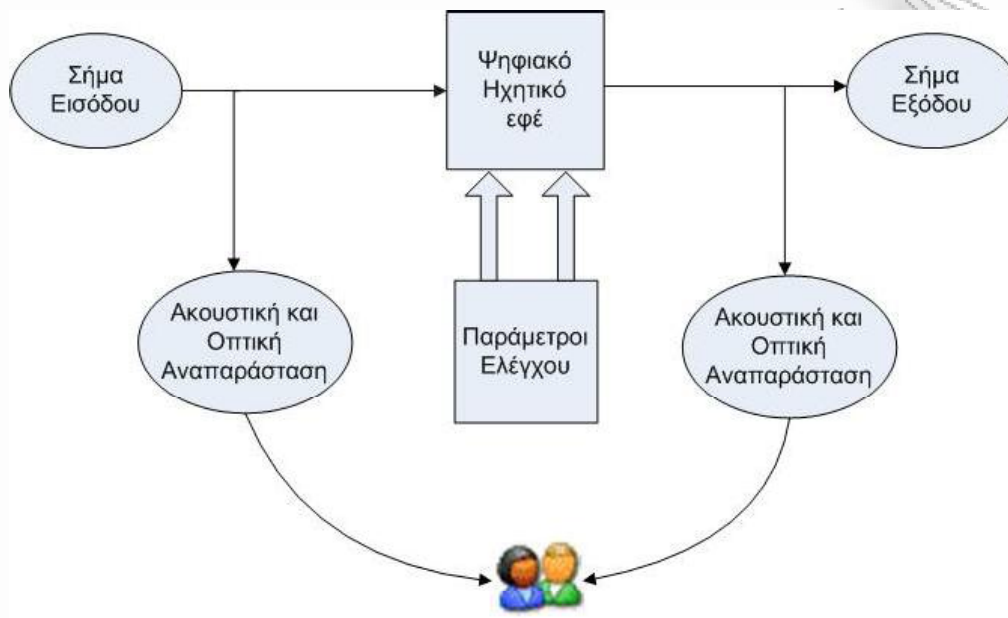
Το να αναπτύξει κάποιος ένα λογισμικό για έναν προσωπικό υπολογιστή (PC) που να υλοποιεί ένα εφέ τύπου Delay-Echo, μέσω ενός προγραμματιστικού περιβάλλοντος υψηλού επιπέδου (πχ Matlab), είναι μια σχετικά εύκολη υπόθεση. Το ίδιο ισχύει και για μία γλώσσα προγραμματισμού όπως η C++ ή Java, αφού υπάρχουν έτοιμες κλάσεις που μπορούν να βοηθήσουν τον προγραμματιστή σε πολύ μεγάλο βαθμό, με συνέπεια να έχει οια χειρά του ένα πρωτότυπο σε λίγο χρόνο. Επιπλέον ένας προσωπικός υπολογιστής έχει πλέον ενσωματωμένο όλο το hardware που απαιτείται για σήματα ήχου: τα κυκλώματα που απαιτούνται για την μετατροπή ενός σήματος από αναλογική σε ψηφιακή μορφή και αντιστρόφως, υπάρχουν μέσα στην κάρτα ήχου, όπως επίσης είναι διαθέσιμη και άφθονη μνήμη RAM. Το κυριότερο όμως είναι ότι έχουμε στην διάθεσή μας έναν επεξεργαστή, ο οποίος στις μέρες μας λειτουργεί σε ταχύτητες μερικών GHz και ο γρήγορος πρόξευ με αριθμούς κινήσης υλοποίησης είναι κάτι παραπάνω από δεδομένες.¹

Έχοντας λοιπόν ως πλατφόρμα προγραμματισμού και υλοποίησης έναν προσωπικό υπολογιστή, πολλές δυνατότητες και ακόμα περισσότερα εργαλεία ανάπτυξης είναι διαθέσιμα. Ακόμη και η αλληλεπίδραση με τον χρήστη για τον έλεγχο των παραμέτρων του λογισμικού είναι δεδομένη μέσω του πληκτρολογίου και του ποντικιού, όπως επίσης είναι εύκολως εφικτή και η δυνατότητα οπτικής αναπαράστασης των σημάτων εισόδου-εξόδου (είτε στο πεδίο του χρόνου είτε στο πεδίο των συχνότητων), μέσω της οθόνης και της κάρτας γραφικών. Τα πράγματα γίνονται λίγο πιο δύσκολα αν θελήσουμε το εφέ που θα φτιάξουμε να λειτουργεί σε πραγματικό χρόνο (Real Time Effect). Και σε αυτήν όμως την περίπτωση έχουν αναπτυχθεί γρήγοροι οδηγοί (Sound card drivers), με τους οποίους η καθυστέρηση μπορεί να φτάσει μέχρι και τα 5ms χωρίς αλλοιώσεις στον ήχο.²

Προφανώς το να σχεδιάσει και να υλοποιήσει κάποιος ένα τέτοιο σύστημα σε ενσωματωμένη μορφή (όπως είναι για παράδειγμα το Digital Delay της Boss) απαιτεί μια τελείως διαφορετική προσέγγιση, λόγω του ότι

¹ Από τότε που εμφανίστηκε ο επεξεργαστής της Intel i860X, ο μαθηματικός πολλαπλασιασμός είναι ενσωματωμένος στην CPU.

² Ένας τέτοιος οδηγός, είναι ο Asio driver της εταιρίας Steinberg.



Σχήμα 1.1: Μοντέλο ψηφιακού ηχητικού εφέ με σύστημα ελέγχου



Σχήμα 1.2: Το κλασικό Boss Digital Delay

τίποτα σχεδόν από όλα όσα αναφέρθηκαν δεν είναι δεδομένο και εύκολα προσβάσιμο. Επιπλέον αν δεν είναι διαθέσιμο κάποιο hardware development board, καταλαβαίνει κανείς ότι τίθεται και το θέμα του αν το όλο εγχείρημα είναι εφικτό!

Στο πρώτο μέρος της εργασίας αυτής θα αναλύσουμε κάποια θεμελιώδη ψηφιακά φίλτρα μέσω των οποίων είναι δυνατή η δημιουργία μιας γραμμής καθυστέρησης (Delay line). Κατόπιν θα περάσουμε στην έννοια της αντήχησης (Reverb) και στο πως μπορούμε να την δημιουργήσουμε σε ένα ψηφιακό σήμα με τεχνητό τρόπο. Το δεύτερο μέρος της εργασίας αφορά την σχεδίαση και την υλοποίηση του συστήματος. Οι απόλυτως αναγκαίοι πόροι για ένα τέτοιο εγχείρημα είναι οι ακόλουθοι:

- Ένας αλγόριθμος υλοποίησης του εφέ
- Ένας προγραμματιζόμενος μικροελεγκτής που θα εκτελέσει τον αλγόριθμο
- Μνήμη RAM
- ADC (Analog to Digital Converter) και DAC (Digital to Analog Converter)
- Ένας τρόπος καθορισμού των παραμέτρων ελέγχου από τον χρήστη
- Ένα περιβάλλον προγραμματισμού (hardware, software)

Το σύστημα μας βασίστηκε στον προγραμματιζόμενο μικροελεγκτή *Proton* της εταιρίας *Maxim*. Ο μικροελεγκτής αυτός περιέχει οκτώ παράλληλους επεξεργαστές, με αποτέλεσμα ο τρόπος προγραμματισμού του να είναι τελείως διαφορετικός από αυτόν που εφαρμόζεται σε άλλους συνηθισμένους μικροελεγκτές. Επιπλέον, η απουσία ενσωματωμένης αριθμητικής μονάδας για πράξεις με αριθμούς κινητής υποδιαστολής (FPU), μας υποχρέωσε να χρησιμοποιήσουμε τεχνικές προγραμματισμού που ενδείκνυνται σε τέτοιες περιπτώσεις (πχ scaling). Αποτέλεσμα του γεγονότος αυτού ήταν η μείωση της πολυπλοκότητας του χρησιμοποιούμενου hardware και κατά συνέπεια η μείωση του κόστους, από την στιγμή που αποφύγαμε τον πειρασμό να προσθέσουμε ένα FPU chip. Τέλος να σημειώσουμε πως τόσο το πρωτότυπο όσο και η πλατφόρμα προγραμματισμού του, κατασκευάστηκαν από εμάς χρησιμοποιώντας τον ελάχιστο αναγκαίο εξοπλισμό, προκειμένου το κόστος του project να κρατηθεί σε χαμηλά επίπεδα.

Κεφάλαιο 2

Γραμμές Καθυστερήσης (Delay Lines)

2.1 Εισαγωγή

Από την Ακουστική γνωρίζουμε ότι ο ήχος ανακλάται όταν προσπίπτει πάνω σε μια επιφάνεια. Εάν η επιφάνεια είναι μακριά από την πηγή και σε απόσταση μεγαλύτερη των 17 μέτρων (πχ πλαγιά βουνού), θα αντιληφθούμε ξέχωρα τον υπό ανάκλαση προερχόμενο ήχο (ηχώ). Εάν όμως είναι κοντά μας, θα καταλάβουμε ίσως κάποιο χρωματισμό με ενίσχυση του αρχικού ήχου χωρίς να αντιληφθούμε κάποια καθυστέρηση (Αντήηση). Όταν τώρα η ηχητική πηγή βρίσκεται μέσα σε έναν κλειστό χώρο, οι επαναλαμβανόμενες ανακλάσεις στους τοίχους δημιουργούν στάσιμα κύματα, "φέρουν" όμως και επιπλέον πληροφορίες τόσο για την δομή των υλικών όσο και για το σχήμα του χώρου. Προφανώς οι αποστάσεις των σημειακών ορίων του χώρου αυτού, καθορίζουν και το αν θα γίνει αντιληπτή η καθυστέρηση σε κάθε ανακλώμενο ηχητικό κύμα. Προκειμένου να εξομοιώσουμε τέτοια ακουστικά φαινόμενα χρειαζόμαστε μια πρωταρχική "δομή" η οποία δεν είναι άλλη από την γραμμή καθυστέρησης (Delay line).

Η γραμμή καθυστέρησης αποτελεί μια απλούστατη δομή μέσω της οποίας μπορούμε να πάρουμε ένα ακουστικό σήμα και να το αναπαράγουμε μετά από κάποιο χρονικό διάστημα (delay time). Ο χρόνος αυτός μπορεί να είναι και λίγα χιλιοστά και δευτερόλεπτα μέχρι μερικά δευτερόλεπτα. Εάν αναμιχθούμε με κάποιο τρόπο το αρχικό μας σήμα με αυτό που έχει καθυστερήσει, τότε - ανάλογα με τον χρόνο καθυστέρησης - μπορούμε να δημιουργήσουμε απλή ηχώ ή ακόμα και να τροποποιήσουμε την χροιά του αρχικού ήχου δημιουργώντας ενδιαφέροντα εφέ (πχ Doubling effect).

Στην επόμενη ενότητα θα διαπιστώσουμε πως ακόμα και μια τόσο θεμελιώδης και απλή δομή, ανάλογα με τον τρόπο χρήσης της, μπορεί να επιφέρει δραστικές αλλαγές στο αρχικό σήμα, τόσο στο πεδίο του χρόνου όσο και στο πεδίο των συχνοτήτων.

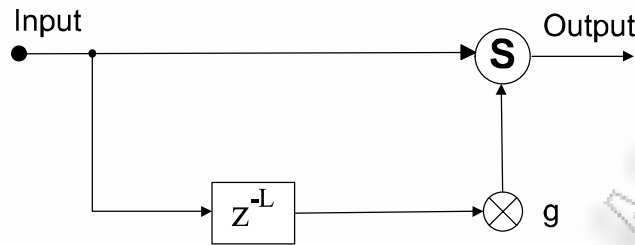
2.2 Ψηφιακά Φίλτρα

Στην ενότητα αυτή θα ασχοληθούμε με φίλτρα τα οποία υλοποιούνται σε υπολογιστές, δηλαδή με Ψηφιακά φίλτρα. Έτσι εδώ ο χρόνος δεν είναι ένα συνεχές μέγεθος που μετρείται σε δευτερόλεπτα αλλά ένα διακριτό μέγεθος που μετρείται σε δείγματα (samples). Αν θέλουμε να μεταβούμε σε πραγματικό χρόνο (sec) δεν έχουμε παρά να πολλαπλασιάσουμε τον αριθμό δείγματος t με την περίοδο δειγματοληψίας T_s . Για διευκόλυνσή μας μπορούμε να εκμεταλλευτούμε το γεγονός αυτό και να μετράμε την κυκλική συχνότητα ω όχι σε rad/sec αλλά σε $rad/sample$. Συνεπώς αυτό είναι η κυκλική συχνότητα δειγματοληψίας να ισούται προς $\omega_s = 2\pi$, ενώ η συχνότητα Nyquist να ισούται με $\omega_N = \pi$. Έτσι ένα phase στον ψηφιακό κόσμο μπορούμε να το γράψουμε με τον ίδιο τρόπο σαν να ήταν αναλογικό σήμα, δηλαδή ως $x_t = e^{j\omega t}$, με την διαφορά πως το ω είναι σε $rad/sample$ και $t \in \mathbb{N}$, ο αριθμός δείγματος. Σύμφωνα με τις συμβάσεις που κάναμε, εάν θέλουμε να καθυστερήσουμε ένα phase για χρόνο ίσο προς ένα δείγμα, αρκεί να το πολλαπλασιάσουμε με την σταθερή μιγαδική ποσότητα $e^{-j\omega}$ (αντί για $e^{j\omega T_s}$) [13].

Στις επόμενες υπό-ενότητες θα αναλύσουμε κάποια βασικά φίλτρα μέσω των οποίων μπορούμε να δημιουργήσουμε δομές καθυστέρησης.

2.2.1 Feedforward Comb Filter

Ας εξετάσουμε το φίλτρο η λειτουργία του οποίου φαίνεται στο σχήμα 2.1. Το αρχικό μας σήμα αφού καθυστερήσει κατά L δείγματα (που αντιστοιχούν σε χρόνο τ sec), πολλαπλασιάζεται με μια πραγματική



Σχήμα 2.1: Διάγραμμα ροής σήματος για ένα απλό Feedforward φίλτρο.

σταθερά $-1 < g < 1$ και προστίθεται στο αρχικό σήμα. Η εξίσωση που ανταποκρίνεται στο απλό αυτό φίλτρο είναι λοιπόν:

$$y_t = x_t + gx_{t-L} \quad \text{και} \quad \tau = L/f_s, \text{ ο χρόνος καθυστέρησης σε sec}$$

Αν συμβολίσουμε με X το σήμα εισόδου που παίρνει τιμές x_t , και με Y το σήμα εξόδου που παίρνει τιμές y_t , η παραπάνω εξίσωση γράφεται ισοδύναμα,

$$Y = X + gz^{-L}X = (1 + gz^{-L})X, \quad \text{όπου } z = e^{j\omega}$$

Προφανώς η συνάρτηση μεταφοράς δίνεται από την έκφραση,

$$H(z) = 1 + gz^{-L}$$

Έστω τώρα πως $g < 0$. Για διευκόλυνσή μας γράφουμε το g ως εξής: $g = -R^L$, όπου $R > 0$. Το φίλτρο αυτό ονομάζεται "Feedforward Comb Filter" και η εξίσωσή του το χαρακτηρίζει όπως και η ονομασία με ακρίβεια του δίνονται από τις σχέσεις:

$$y_t = x_t - R^L x_{t-L} \quad (2.1)$$

$$H(z) = 1 - R^L z^{-L} = \frac{z^L - R^L}{z^L} \quad (2.2)$$

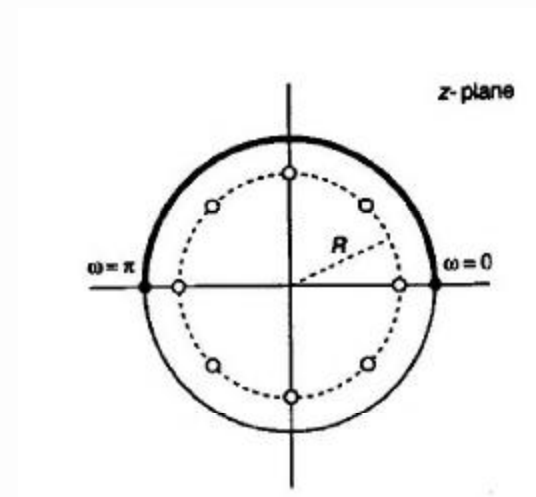
Ως γνωστόν ο γεωμετρικός τόπος των ριζών της εξίσωσης 2.2 είναι κύκλος ακτίνας R , οι δε ρίζες βρίσκονται σε ίσες αποστάσεις μεταξύ τους και είναι L το πλήθος. Ισχύει λοιπόν,

$$z_k = R e^{jk2\pi/L}, \quad \text{όπου } k = 0, \dots, L-1$$

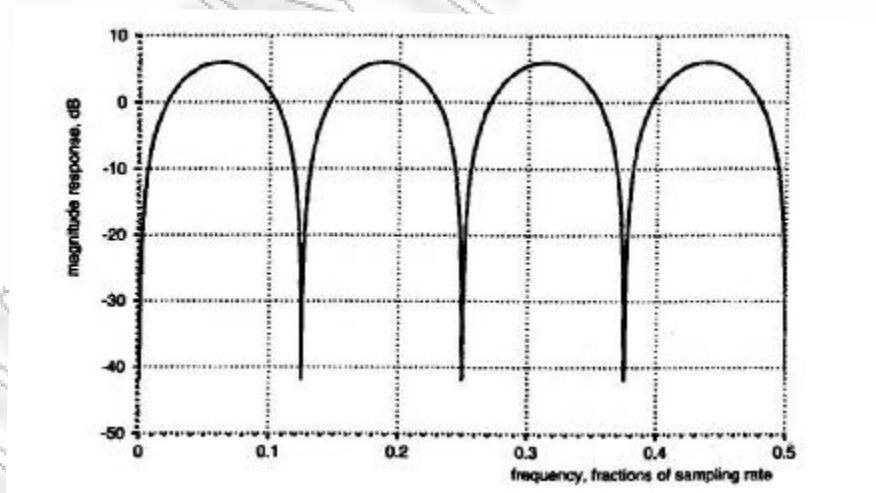
Στο σχήμα 2.2 [13], φαίνεται μια περίπτωση όπου το αρχικό σήμα καθυστερεί κατά 8 δείγματα, οπότε έχουμε 8 μηδενικά (zeros) στο μιγαδικό επίπεδο. Προφανώς όσο το R πλησιάζει την μονάδα, τόσο πιο κοντά στον μοναδιαίο κύκλο θα είναι τα μηδενικά, και άρα το μέτρο της απόκρισης συχνοτήτων του φίλτρου θα είναι πολύ μικρό για συχνότητες γετονικές με αυτά τα σημεία μηδενισμού της συνάρτησης μεταφοράς του. Με άλλα λόγια για τις συχνότητες $f_k = 2kf_N/L$, (όπου $k = 0, \dots, L/2$ και $f_N = f_s/2$ η συχνότητα Nyquist), το μέτρο της απόκρισης του φίλτρου θα είναι κοντά στο 0 (με δεδομένο ότι το R είναι πολύ κοντά στο 1). Το γεγονός αυτό φαίνεται στο σχήμα 2.3 [13], όπου μπορεί κάποιος να καταλάβει και τον λόγο για τον οποίο το φίλτρο αυτό ονομάζεται "Comb Filter", αφού η απόκριση συχνοτήτων του έχει την μορφή χτένας. Στο παράδειγμα αυτό επειδή το $R=0.999$, η απόκριση συχνοτήτων του φίλτρου είναι -12 db για τις συχνότητες $f_k = kf_N/4$ με $k = 0, 1, 2, 3, 4$, έχουμε δηλαδή εξασθένιση κατά 125 φορές ($|H(\omega_k)| = 0.008$).

Αν θέλουμε να εξετάσουμε το μέτρο της απόκρισης συχνοτήτων του φίλτρου αυτού, σε σχέση με τις παραμέτρους τ, g , μπορούμε να διακρίνουμε δύο περιπτώσεις ως προς το πρόσημο του g .

- Εάν το $g > 0$, το φίλτρο εξασθενεί με τον παράγοντα $1 - |g|$ όλες τις συχνότητες $f_k = kf_N/L$, όπου k περιττός.
- Εάν το $g < 0$, το φίλτρο εξασθενεί με τον παράγοντα $1 + |g|$ όλες τις συχνότητες που είναι ακέραια πολλαπλάσια του $1/\tau$, δηλαδή τις συχνότητες $f_k = kf_s/L$, όπου $k \in \mathbb{N}$.
- Και στις δύο περιπτώσεις το - εξαρτώμενο από την συχνότητα - κέρδος του φίλτρου, βρίσκεται ανέμεσο στις τιμές $1 - |g|$ και $1 + |g|$. Τοίσι σημαίνει πως η μεγαλύτερη δυνατή ενίσχυση που μπορεί να έχουμε για $|g| < 1$, είναι δύο φορές, δηλαδή 6 db.



Σχήμα 2.2: Οι οκτώ μιγαδικές ρίζες ενός Feedforward Comb φίλτρου, ισαπέχουν μεταξύ τους και βρίσκονται σε κύκλο ακτίνας $R < 1$

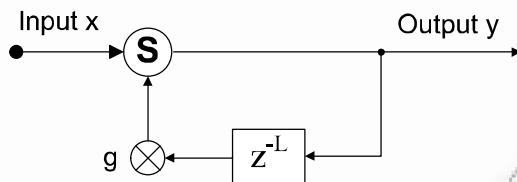


Σχήμα 2.3: Απόκριση συχνοτήτων ενός Feedforward Comb φίλτρου με 8 μιγαδικές ρίζες, στην περίπτωση που $R=0.999$

Μετά από την προηγούμενη ανάλυση, διαπιστώνουμε πως η επίδραση του φίλτρου αυτού στο αρχικό σήμα είναι εξίσου σημαντική τόσο στο πεδίο του χρόνου, όσο και στο πεδίο των συχνοτήτων. Ανάλογα με το μέγεθος της καθυστέρησης L , μπορούμε να εκτιμήσουμε την επίδραση του φίλτρου εξελέγοντας την μια ή την άλλη αναπαράσταση. Αν δηλαδή το L είναι λίγα δείγματα, προφανώς δεν μπορούμε να καταλάβουμε κάποια καθυστέρηση στο σήμα, δεδομένου ότι το αυτό μπορεί να διακρίνει δύο ήχους μόνο αν απέχουν χρονικά τουλάχιστον $1/10$ sec. Έχουμε όμως αισθητή διαφορά στην χροιά του ήχου, αφού όπως είδαμε αλλάζει δραστικά το περιεχόμενο των συχνοτήτων του σήματος. Από την άλλη μεριά, αν το L είναι μεγάλο, το ανθρώπινο αυτί καταλαβαίνει την διαφορά που συντελείται στο πεδίο του χρόνου, ως καθυστέρηση του αρχικού σήματος (echo), ενώ δεν μπορεί να διακρίνει την αλλαγή στο πεδίο των συχνοτήτων, αφού οι συχνότητες που εξασθενούν είναι στην περίπτωση αυτή πολύ κοντά μεταξύ τους.

Να σημειώσουμε πως στην ακραία περίπτωση που $L=1$ δείγμα, το φίλτρο μας για $g > 0$ είναι ένα βαθυπερατό φίλτρο (LPE), ενώ για $g < 0$ είναι ένα υπεραπαστό φίλτρο (HPE). Τέλος να αναφέρουμε πως το φίλτρο αυτό ανήκει στην κατηγορία των FIR (Finite Impulse Response) φίλτρων.

2.2.2 Feedback Comb Filter



Σχήμα 2.1: Διάγραμμα ροής σήματος για ένα Feedback Comb φίλτρο.

Όπως είδαμε στην προηγούμενη ενότητα το Feedback Comb φίλτρο περιγράφεται από την εξίσωση 2.1. Το φίλτρο αυτό χρησιμοποιεί παρελθόντα δείγματα του σήματος εισόδου x_{t-L} , και για αυτόν τον λόγο ανήκει στην κατηγορία των feedforward φίλτρων. Ας θεωρήσουμε τώρα ένα φίλτρο σαν το προηγούμενο με την διαφορά όμως ότι αυτό θα χρησιμοποιεί παρελθόντα δείγματα του σήματος εξόδου y_{t-L} αντί για x_{t-L} , όπως φαίνεται στο σχήμα 2.4. Πατάω ότι $g = R^L > 0$. Η εξίσωση που περιγράφει το φίλτρο μας θα είναι,

$$y_t = x_t + R^L y_{t-L} \quad (2.3)$$

Το φίλτρο αυτό το ονομάζουμε "Feedback Comb" και είναι ένα φίλτρο με ανάδραση (feedback filter). Η συνάρτησι μεταφοράς του θα είναι:

$$H(z) = \frac{1}{1 - R^L z^{-L}} = \frac{z^L}{z^L - R^L} \quad (2.4)$$

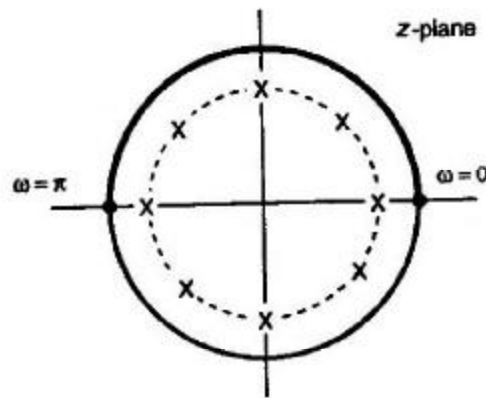
Παρατηρώντας τις σχέσεις 2.2 και 2.4, παρατηρούμε πως η συνάρτησι μεταφοράς αυτού του (feedback) φίλτρου είναι η αντίστροφη αυτής που χαρακτηρίζει το (feedforward) φίλτρο που εξετάσαμε στην προηγούμενη ενότητα. Συνεπώς το ίδιο ακριβώς θα ισχύει και για το μέτρο της απόκρισης συχνοτήτων του. Συνέπεια αυτού, είναι το γεγονός πως αν ένα σήμα περάσει διαδοχικά από τα δύο αυτά φίλτρα (με οποιαδήποτε σειρά), θα μείνει άθικτο, αφού τα δύο φίλτρα θα αλληλοεξουδετερωθούν [13].

Στο σχήμα 2.5 [13], βλέπουμε τους πόλους του φίλτρου για $L = 8$, οι οποίοι ως γνωστόν αποτελούν τις ρίζες του παρανομαστή της συνάρτησις μεταφοράς (εξίσωση 2.4). Είναι φανερό πως αυτοί βρίσκονται ακριβώς στις ίδιες θέσεις με τα ζεύγη του Feedforward comb φίλτρου της προηγούμενης ενότητας (σχήμα 2.2), δηλαδή πάνω σε κύκλο ακτίνας R .

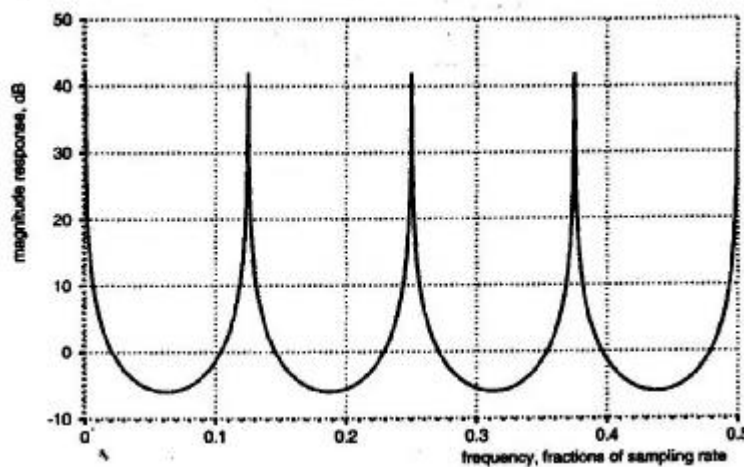
Όσον αφορά την γραφική παράσταση του μέτρου της απόκρισης συχνοτήτων, είναι η αντίστροφη αυτής που χαρακτηρίζει το "Feedforward Comb" φίλτρο της προηγούμενης ενότητας, όπως φαίνεται στο σχήμα 2.6 [13], για $L = 8$. Με άλλα λόγια αν περιστρέψουμε κατά 180° το σχήμα 2.3, θα πάρουμε την γραφική παράσταση του σχήματος 2.6. Έτσι τώρα η γραφική παράσταση του μέτρου της απόκρισης συχνοτήτων, αποτελείται από "εξογκώματα" (resonance peaks) των οποίων η οξύτητά τους είναι τόσο μεγαλύτερη όσο το $|g| > 1$. Κάθε ένα τέτοιο εξογκώμα αντιστοιχεί σε ένα ζεύγρι συζυγών μιγαδικών πόλων.

Στο σημείο αυτό πρέπει να αναφερθεί πως το "Comb" φίλτρο που εξετάζουμε, έχει κρουστική απόκριση (Impulse Response) με άπειρους όρους. Το φίλτρο δηλαδή αυτό σε αντιδιαστολή με εκείνο της προηγούμενης υπενόησης, ανήκει στην κατηγορία των IIR (Infinite Impulse Response) φίλτρων (βλ. σχήμα 2.7). Είναι πολύ εύκολο να δει κανείς πως προκειμένου το φίλτρο να είναι ευσταθές θα πρέπει να ισχύει η σχέση $|g| \leq 1$, ειδικάως το σήμα εξόδου θα μεγαλώνει συνεχώς χωρίς όριο.

Εξετάζοντας την κρουστική απόκριση του φίλτρου, βλέπουμε ότι αυτή ομοιάζει με την κρουστική απόκριση που θα παίρναμε για έναν χώρο μεταξύ δύο τοίχων, όπου το ακουστικό κύμα γίνεται ολοένα και πιο αδύναμο μετά από κάθε ανάκλαση. Η καθυστέρηση οπτικά εκφράζει τον χρόνο που κάνει το κρουστικό κύμα μέχρι να επιστρέψει ξανά στην πηγή, μετά από δύο ανακλάσεις (round trip delay). Το κέρδος g καθορίζει τον



Σχήμα 2.5: Οι οκτώ πόλοι ενός Feedback Comb φίλτρου ($g > 0$), ισαπέχουν μεταξύ τους και βρίσκονται σε κύκλο ακτίνας $R < 1$



Σχήμα 2.6: Απόκριση ούχνο-ήταν ενός Feedback Comb φίλτρου με 8 πόλους, στην περίπτωση που $R = 0.999$ και $g > 0$

χρόνο αντήχησης (βλ. ενότητα 3.1). Αν θέλουμε να είμαστε περισσότερο ακριβείς μπορούμε να πούμε πως το "Feedback Comb" φίλτρο με κέρδος $g > 0$ και καθυστέρηση L δείγματα, εξομοιώνει την ηχητική συμπεριφορά ενός κυλινδρικού σωλήνα, όπου και τα δύο του άκρα είναι είτε κλειστά, είτε ανοικτά [13].

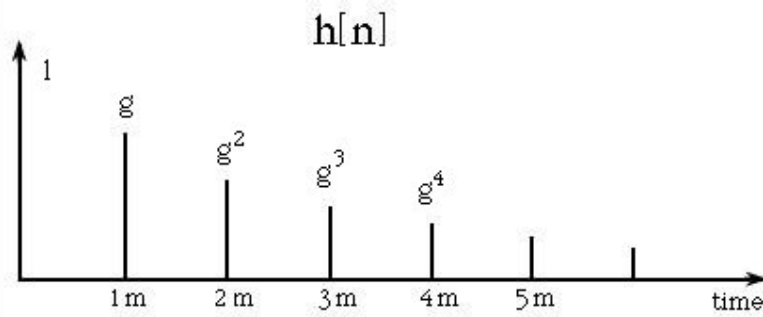
Σύμφωνα με την ανάλυση που κάναμε είναι προφανές πως το - εξαρτώμενο από την συχνότητα - κέρδος του φίλτρου βρίσκεται μεταξύ των τιμών $\frac{1}{1+|g|}$ και $\frac{1}{1-|g|}$, οπότε η ενίσχυση του σήματος είναι πολύ μεγάλη όσο το g πλησιάζει προς το 1.¹ Προκειμένου να αντιμετωπιστεί το φαινόμενο αυτό, για περιοδικά σήματα, μπορούμε να ελασθενήσουμε το σήμα εισόδου με τον παράγοντα $c = 1 - |g|$, εξασφαλίζοντας ότι δεν θα συμβεί υπερφόρτωση [14].

Ας εξετάσουμε τώρα την περίπτωση που το $g = -R^L < 0$. Η συνάρτηση μεταφοράς του φίλτρου θα είναι,

$$H(z) = \frac{1}{1 + R^L z^{-L}} = \frac{z^L}{z^L + R^L}$$

Οι πόλοι αυτού του φίλτρου είναι ίδιοι με αυτούς που θα παίρναμε για $g > 0$, αν τους μετατοπίσουμε κατά γωνία π/L πάνω στον μιγαδικό κύκλο ακτίνας R . Επίσης οι πόλοι αυτοί είναι όλοι περιττές αρμονικές της θεμελιώδους συχνότητας π/L . Να σημειώσουμε πως ένα τέτοιο φίλτρο - με αρνητικό κέρδος ανάδρασης και με καθυστέρηση L δείγματα - εξομοιώνει έναν κυλινδρικό σωλήνα, με ανοικτό το ένα του άκρο και κλειστό το

¹ Είναι προφανές ότι αυτό ισχύει για μικρές τιμές του L , όπου το ανθροστικό ασθθητήριο της ακούς δεν μπορεί αντιληφθεί τις αλλαγές στο πεδίο του χρόνου.

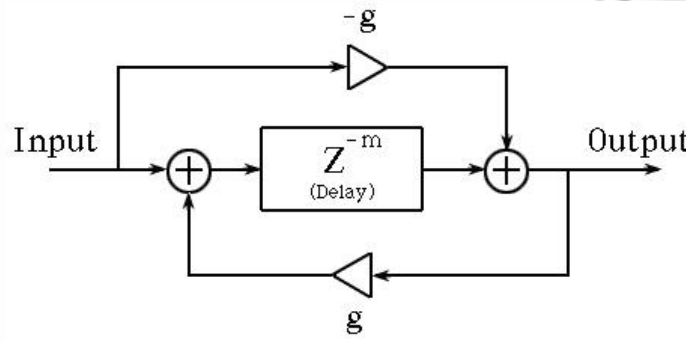


Σχήμα 2.7: Κρουστική Απόκριση ενός Feedback Comb φίλτρου. Το feedback gain είναι g και έχουμε καθυστέρηση κατά m δείγματα.

έλλο. Το γεγονός ότι στην περίπτωση αυτή έχουμε μόνο πεπεσμένες αρμονικές μιας θεμελιώδους συχνότητας, εξηγεί και τον ιδιαίτερο ήχο των πνευστών οργάνων [13].

2.2.3 All-Pass Filter

Ένα φίλτρο All-Pass έχει την μοναδική ιδιότητα να διατηρεί ανέπαφο (σταθερό) το μέτρο της απόκρισης συχνότητων του για όλες τις συχνότητες, οπότε όλες οι συχνότητες περνούν από αυτό με το ίδιο βάρος. Αποτέλεσμα αυτού είναι να μην υπάρχει "χρωματισμός" του σήματος.^{*} Η εξίσωση που δίνει ένα τέτοιο φίλτρο είναι η, $y_t = gx_t + x_{t-m} + gy_{t-m}$, όπου m είναι η καθυστέρηση σε αριθμό δειγμάτων. Στο σχήμα 2.8 βλέπουμε το διάγραμμα ροής σήματος για ένα All-Pass φίλτρο, όπου είναι φανερό η ομοιότητα με ένα Feedback Comb φίλτρο στο οποίο έχουμε προσθέσει ένα feedforward path με αρνητικό gain $-g$:



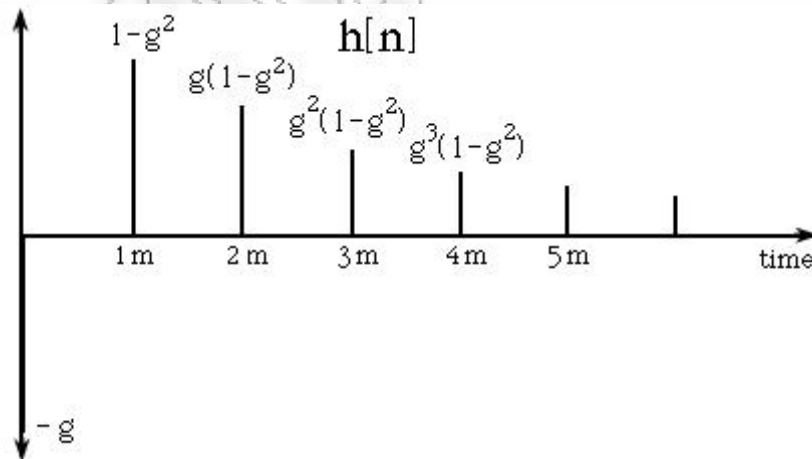
Σχήμα 2.8: Διάγραμμα ροής σήματος για ένα All-Pass φίλτρο.

Στο σχήμα 2.9 φαίνεται η κρουστική απόκριση ενός All-Pass φίλτρου. Απ' αυτήν είναι φανερό πως το φίλτρο απλά είναι διαιτηδόν να χρησιμοποιηθεί για να εξισορροπούμε τον χρόνο που εβήσαν οι ανακλίσεις σε ένα δωμάτιο, κατά την υλοποίηση ενός συστήματος τεχνητής αντήχησης.

Η συνάρτηση μεταφοράς ενός All-Pass φίλτρου πρώτης τάξης είναι η,

$$H(z) = \frac{1 - \bar{a}z}{z - a}$$

,όπου $\bar{a} \in \mathbb{C}$, ο συζυγής μιγαδικός του $a \in \mathbb{C}$. Βλέπουμε λοιπόν πως έχουμε έναν πόλο στην θέση $z = a$ και ένα zero στην θέση $z = 1/\bar{a}$. Αυτή η σχέση που υπάρχει μεταξύ του πόλου και του zero είναι και η αιτία που περνούν όλες οι συχνότητες ανέπαφες.



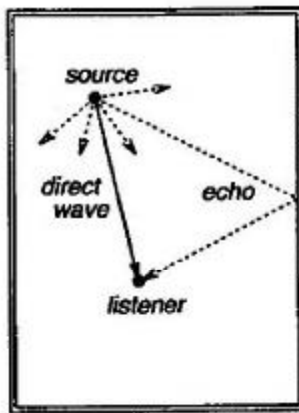
Σχήμα 2.9: Κρουστική απόκριση για ένα All-Pass φίλτρο πρώτης τάξης. Η καθυστέρηση είναι ίση προς m δείγματα.

^{*}Το ίδιο ισχύει εάν η καθυστέρηση είναι σημαντικά μικρότερη από τον χρόνο ολοκλήρωσης του αιτιού, περίπου $\leq ms$ [14]

Κεφάλαιο 3

Αντήχηση (Reverb)

3.1 Φυσική Αντήχηση



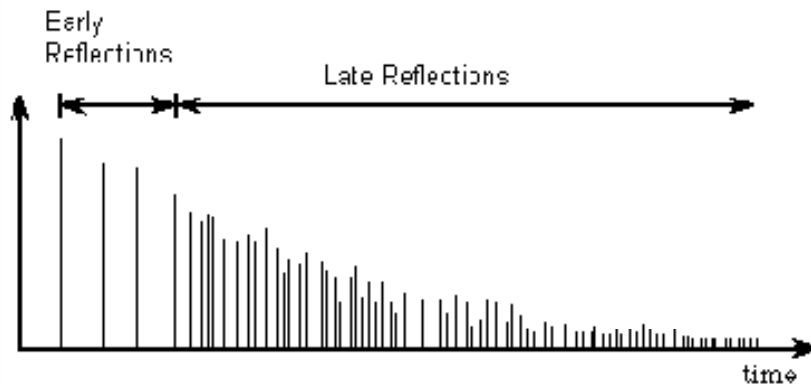
Σχήμα 3.1: Το φαινόμενο της φυσικής αντήχησης σε έναν κλειστό χώρο, προκαλείται από την υπέρθεση πολλών ανακλώμενων κυμάτων. Εδώ φαίνεται μόνον ένα ανακλώμενο κύμα.

να εξαπλώνονται, να ανακλώνται και να απορροφώνται με αξιημένο ρυθμό, ο οποίος συνεχώς μεταβάλλεται. Αυτό γίνεται άπειρες φορές και πλέον δεν ξεχωρίζουμε μεμονωμένες ανακλάσεις, αλλά μια συμπαγή και ομοιογενή αντήχηση, η οποία σταδιακά χάνει την ενέργειά της και σβήνει. Πρέπει να σημειωθεί πως αυτές οι "συμπαγείς" ανακλάσεις (Late Reflections) είναι περισσότερο τυχαίες και σχετίζονται με πολύπλοκο τρόπο με τα φυσικά χαρακτηριστικά του χώρου. Ο χρόνος που περνάει μέχρι η έντασή της ομοιογενούς αυτής αντήχησης να μειωθεί κατά 1/1000 (60 db) ονομάζεται Χρόνος Αντήχησης (Reverberation time T_{60}). Ενδεικτικά αναφέρουμε μερικές τιμές για γεμάτες αίθουσες:[1]

- Symphonic Hall (Βοστώνη) : 1.8 sec
- Ουλαρμονική Βερολίνου : 1.9 sec
- Σκάλο Μάιναμ : 1.2 sec

Μια άδεια αίθουσα έχει σαφώς μικρότερο συντελεστή απορρόφησης και οι παραπάνω τιμές σχεδόν διπλασιάζονται. Πάντως ένας μέσος χώρος έχει χρόνο αντήχησης που ξεκινάει από λίγα δεκάτα του δευτερόλεπτου και φτάνει περίπου στο δευτερόλεπτο.

Impulse Response



Σχήμα 3.2: Κρουστική απόκριση ενός κλειστού χώρου

Μέχρι στιγμής εξετάσαμε την βασική δυναμική συμπεριφορά της αντήχησης, που μας δίνει ακουστικές πληροφορίες για το μέγεθος του χώρου. Εμείς όμως, πέρα από το μέγεθος του χώρου θέλουμε να ξέρουμε και εάν ένας χώρος έχει ήχο μουντό ή λαμπερό (ανάλογα με τα υλικά από τα οποία είναι χτισμένος) και αν είναι πολύπλοκα δομημένος. Η αντήχηση μας δίνει αυτές τις πληροφορίες ανάλογα με το πόσο πυκνή είναι και το πόσο γρήγορα εξασθενούν οι υψηλές συχνότητες και κατά επέκταση τα επιμέρους φάσματα. Έτσι υλικά όπως μπετόν, τούβλα, τζάκια, και πλακάκια ανακλούν έντονα τις υψηλές συχνότητες με αποτέλεσμα να έχουμε αύξηση του χρόνου αντήχησης. Αντιθέτως "αντικείμενα" με μεγάλο συντελεστή απορρόφησης, όπως κουρτίνες, χαλιά ή άνθρωποι, μειώνουν τον χρόνο αντήχησης. Επίσης δωμάτια μεγάλων διαστάσεων έχουν την τάση για μεγάλους χρόνους αντήχησης, αφού κατά μέσον όρο τα ηχητικά κύματα διανύουν μεγαλύτερες αποστάσεις μεταξύ διαδοχικών ανακλάσεων. Τέλος να σημειώσουμε πως ο αέρας σε ένα δωμάτιο επίσης συντελεί στην εξασθένηση των ηχητικών κυμάτων, μειώνοντας τον χρόνο αντήχησης. Οι συχνότητες που πληττύνται περισσότερο είναι οι υψηλές, γεγονός που μας οδηγεί στην χρήση βαθυπερατών φίλτρων κατά την υλοποίηση ενός συστήματος τεχνητής αντήχησης.

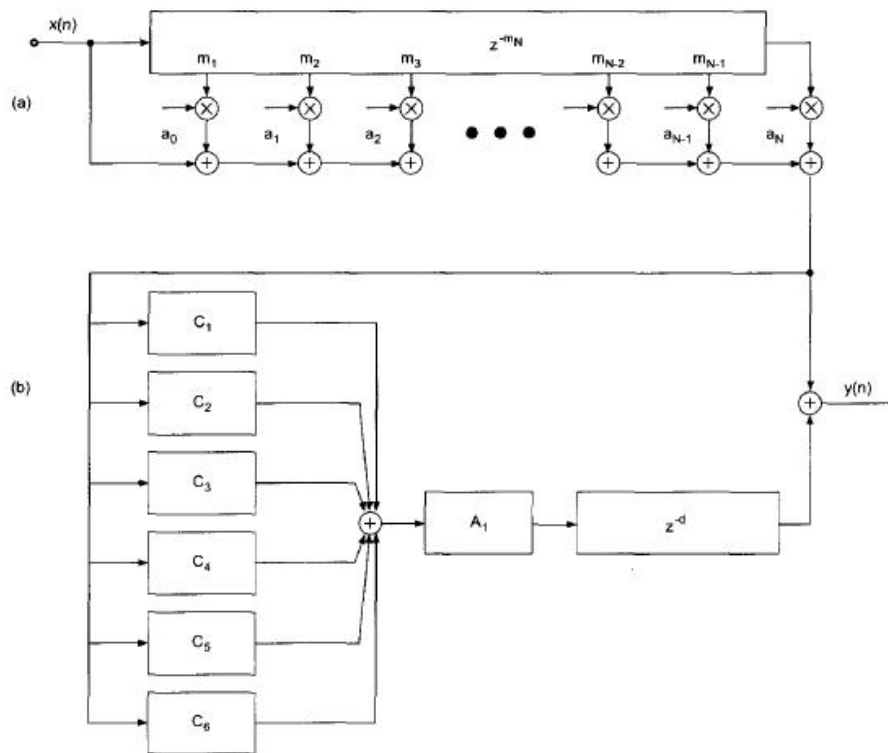
3.2 Ψηφιακά συστήματα Τεχνητής Αντήχησης

Η βασική δομή στην οποία στηρίζεται ένα ψηφιακό σύστημα τεχνητής αντήχησης είναι το Feedback Comb φίλτρο που εξετάσαμε αναλυτικά στην υποενότητα 2.2.2. Πράγματι, αν κάνουμε την παραδοχή ότι η ηχώ αποτελείται από επαναλαμβανόμενες "εκδόσεις" του αρχικού ήχου, κάθε μια από τις οποίες φθάνει στον ακροατή μετά από καθορισμένο σταθερό χρόνο σε σχέση με την προηγούμενη, και επιπλέον εξασθενημένη ως προς το πλάτος της κάθε φορά κατά $g < 1$. Αν δεχτούμε όλα τα παραπάνω τότε παίρνουμε μια απλοποιημένη μορφή ενός συστήματος τεχνητής αντήχησης, με κρουστική απόκριση ίδια με αυτήν ενός Feedback comb φίλτρου (βλ. σχήμα 2.7).

Πόση όμως πρέπει να είναι η καθυστέρηση (σε milliseconds) για ένα Feedback Comb φίλτρο, προκειμένου να έχουμε έστω μια χοντρική προσομοίωση της συμπεριφοράς ενός συνηθισμένου δωματίου. Κατ' αρχήν, όπως αναφέραμε και στην υποενότητα 2.2.2, στην ουσία εξομοιώνουμε έναν χώρο που περιορίζεται από δύο μόνον τοίχους. Εάν λοιπόν θεωρήσουμε την απόσταση που διανύει ένα ηχητικό κύμα μέχρι να επιστρέψει στην πηγή και οπιστάνομε 2 ανακλάσεις, γύρω στα 20 μέτρα, ο χρόνος που θα απαιτηθεί θα είναι ίσος προς $delay = 20/c$, όπου $c = 344 \text{ m/sec}$, είναι η ταχύτητα διάδοσης του ήχου στον αέρα. Η τιμή που προκύπτει είναι περίπου 60 msec . Η αντίστοιχη θεμελιώδης συχνότητα αντήχησης για ένα τέτοιο "δωμάτιο" θα είναι $f_0 = 1/delay = 17 \text{ Hz}$ [13].

Αν θέλοιμε να δομήσουμε σημεία που υατερεί η χρήση ενός μόνο Feedback Comb φίλτρου, για την εξομοίωση ενός δωματίου, δεν έχουμε παρά να συγκρίνουμε τις κρουστικές αποκρίσεις τους, όπως αυτές δόθηκαν στα σχήματα 3.2 και 2.7. Είναι φανερό πως σε ένα πραγματικό δωμάτιο, ο ρυθμός που επιστρέφουν οι ανακλάσεις δεν είναι σταθερός (όπως στο Feedback Comb) αλλά αλλάζει με την πάροδο του χρόνου.

Τέλος ένα άλλο θέμα έχει να κάνει με την χρονική διαφορά που έχουν τα κύματα που φτάνουν στα δύο αυτιά μας. Πρακτε μένου να αναληφθούμε την αντήχηση θα πρέπει τα κέρματα αυτά να είναι όσο το δυνατόν περισσότερο ανεξάρτητα ή με άλλα λόγια θα πρέπει να ακολουθούν διαφορετικές διαδρομές για το κάθε αυτί,



Σχήμα 3.3: Δομή ενός δημοφιλούς συστήματος τεχνητής αντήχησης, από τον James Moorer.

διανύοντας διαφορετικές αποστάσεις.¹

Από τα παραπάνω είναι προφανές πως ένα απλό Feedback Comb φίλτρο προσεγγίζει πολύ χοντρικά την ηχητική συμπεριφορά ενός δωματίου. Γενικά είναι απαραίτητος ο συνδυασμός πολλών φίλτρων για να έχουμε κάποιο ρεαλιστικό αποτέλεσμα. Από τις μέχρι τώρα προσπάθειες έχει φανεί πως το πρόβλημα αυτό δεν λύνεται αλγοριθμικά, αφού πολλές επιτυχημένες προσπάθειες βασίστηκαν σε πλήθος δοκιμών (trial and error) και σε εμπνεύσεις των ερευνητών, δικαιώνοντας αυτούς που υποστηρίζουν πως η υλοποίηση ενός υψηλής πιστότητας συστήματος τεχνητής αντήχησης, είναι περισσότερο τέχνη παρά επιστήμη.

Μια τέτοια προσπάθεια με πολύ καλά αποτελέσματα περιγράφεται σε ένα πολύ γνωστό άρθρο του James Moorer [8], η οποία συνδυάζει όπως θα δούμε διάφορα Comb και All-Pass φίλτρα για να πετύχει τον στόχο της.

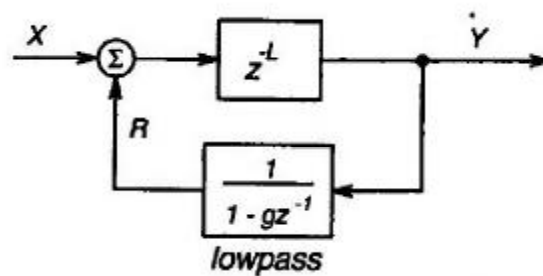
Στο σχήμα 3.3 [14], βλέπουμε την δομή που προτείνει ο James Moorer για ένα σύστημα τεχνητής αντήχησης. Το τμήμα (α) είναι ουσιαστικά μια γραμμή καθυστέρησης με N το πλήθος εξόδους, όπου παίρνουμε σε διάφορα σημεία της, καθυστερημένες εκδόσεις του αρχικού σήματος τις οποίες και αθροίζουμε (tapped delay line). Με τον τρόπο αυτό δημιουργούμε τις πρώτες ανακλάσεις που περιγράψαμε στην προηγούμενη ενότητα και που είδαμε στο σχήμα 3.2. Η έξοδος του τμήματος (α) μεταφράζεται το τμήμα (β) (Diffuse Reverberator) που είναι υπεύθυνο για την δημιουργία των συμπαγών ανακλάσεων (Late Reflections). Να σημειώσουμε πως η γραμμή καθυστέρησης των d δειγμάτων στη έξοδο του τμήματος (β), υπάρχει προκειμένου οι πρώτες ανακλάσεις να φτάσουν χρονικά πριν από τις συμπαγείς ανακλάσεις που παράγει το τμήμα (β).

Η δομή που προτείνει ο Moorer για το diffuse reverberator συνίσταται από έξι διαφορετικά Feedback Comb φίλτρα εν παραλληλία, ακολουθούμενα από ένα All-Pass φίλτρο. Κάθε ένα από τα έξι αυτά Comb φίλτρα περιέχει και ένα απλό (με έναν μόνο πόλο) βαθυπερατό φίλτρο, όπως φαίνεται στο σχήμα 3.4 [14]. Με τον τρόπο αυτό το κέρδος της ανάδρασης κάθε Feedback Comb φίλτρου, δεν είναι το ίδιο για όλες τις συχνότητες, με αποτέλεσμα να εκδεικνύεται ο εγγενής “μεταλλικός” ήχος² του φίλτρου αυτού.

Προκειμένου το σύστημα να λειτουργήσει ορθά, θα πρέπει να ελεγχθούν με προσοχή οι διάφορες παράμετροι

¹ Αυτός είναι και ο λόγος που οι αθήσεις μουσικής είναι ηχηλεπώρες, με αποτέλεσμα οι πρώτες ανακλάσεις να φιλάνουν την ακραστή τα διαφορετικές χρονικές για το κάθε αντί, αφού οι ανακλάσεις αυτές μεταφέρονται από τους παλιούς τοίχους και δεν βραδύνονται πονήως σε ίσες αποστάσεις από την ακραστή.

² Μια εξήγηση για την αίσθηση του “μεταλλικού” ήχου που παράγεται από ένα απλό Comb φίλτρο, είναι το γεγονός της υπερβολικής ενίσχυσης των υψηλών συχνοτήτων της συμπαγούς αντήχησης σε βάρος των χαμηλών.



Σχήμα 3.4: Ένα Comb φίλτρο που περιέχει στο feedback loop ένα απλό βαθυπερατό φίλτρο, προσομοιώνει την εξασθένηση των υψηλών συχνοτήτων που υφίσταται ο ήχος κατά την διάδοσή του στον αέρα.

των φίλτρων που το απαρτίζουν. Ο Μουρτζι λοιπόν στο άρθρο του προτείνει σχετ. κά με το All-Pass φίλτρο, μια καθυστέρηση γύρω στα 6 ms και για παράμετρο a την πραγματική τιμή $a=0.7$. Όσον αφορά τα έξι Comb φίλτρα, οι καθυστερήσεις θα πρέπει να βρίσκονται μέσα στο διάστημα από 30 έως 80 ms, και επιπλέον οι αντίστοιχες τιμές τους σε αριθμό δειγμάτων πρέπει να είναι αμοιβαία πρώτοι αριθμοί (ή να μην διαρούνται μεταξύ τους). Με τον τρόπο αυτό μειώνεται η υπέρθεση των αντηχήσεων στην κρουστική απόκριση του συστήματος. Τέλος η εκλογή των παραμέτρων g και R για κάθε βαθυπερατό φίλτρο (βλ σχήμα 3.4) παίζει πολύ σπουδαίο ρόλο και μια επιπόλαια εκτίμηση μπορεί καταστήσει όλο το σύστημα ασταθές [13]. Στην πράξη οι τιμές για τα R_i καθορίζουν τον χρόνο αντήχησης, T_{60} , ενώ τα κέρδη g_i για κάθε βαθυπερατό φίλτρο, εξαρτώνται από παράγοντες όπως, η συχνότητα δειγματοληψίας, η καθυστέρηση σε δείγματα L , καθώς και η υγρασία, η πίεση και η θερμοκρασία του χώρου που θέλουμε να εξομοιώσουμε.

Μέρος II

Σχεδίαση ενσωματωμένου συστήματος
ψηφιακού Delay-Echo

Κεφάλαιο 4

Μεθοδολογία ανάπτυξης του συστήματος

4.1 Φάσεις Ανάπτυξης

Όπως έχει αναφερθεί στο κεφάλαιο 1, η ανάπτυξη ενός ενσωματωμένου συστήματος (Embedded System) παρουσιάζει κάποιες ιδιαιτερότητες και δυσκολίες σε σχέση με την ανάπτυξη λογισμικού για έναν προσωπικό υπολογιστή. Με δεδομένα την ανεπαρκή υποδομή και πόρων, θελήσαμε να σχεδιάσουμε και να υλοποιήσουμε ένα όσο το δυνατόν απλού σύστημα που να δημιουργεί σε πραγματικό χρόνο το εφέ της καθυστέρησης ενός ακουστικού σήματος, όπως κάνουν πολλά τέτοια συστήματα που κυκλοφορούν στο εμπόριο. Προκειμένου να πετύχουμε τον στόχο μας, ακολουθήσαμε το “Γενικό Μοντέλο Κύκλου Ζωής Λογισμικού” το οποίο συγκροτείται από κύκλους ανάπτυξης. Τυπικά σε κάθε κύκλο ανάπτυξης υπάρχουν οι εργασίες της σύνταξης των Προδιαγραφών - Απαιτήσεων, της Ανάλυσης, της Σχεδίασης, της Υλοποίησης και του Ελέγχου. Στην παρούσα ενότητα δεν θα αναφερθούμε λεπτομερώς σε όλον τον κύκλο εργασιών των διαφόρων Φάσεων του Γενικού Μοντέλου¹, αλλά στα πρώτα δύο κομμάτια του κύκλου ανάπτυξης, δηλαδή στις εργασίες που αφορούν τον καθορισμό των Προδιαγραφών - Απαιτήσεων και σε αυτήν της Ανάλυσης του συστήματος. Σχετικά με τις εργασίες της Σχεδίασης, της Υλοποίησης και του Ελέγχου, θα ασχοληθούμε αναλυτικά στα δύο επόμενα κεφάλαια.

Φάση “Σύλληψης της ιδέας” (Conception).

Αρχικά έπρεπε να τεθούν οι απαιτήσεις-προδιαγραφές του συστήματος μας. Στο σημείο αυτό βοηθήσει η ανάλυση παρόμοιων ενσωματωμένων συστημάτων επεξεργασίας ήχου [10, 9]. Το φίλτρο που θα υλοποιήσουμε είναι το Feedback Comb του σχήματος 2.4. Μια σημαντική προδιαγραφή του συστήματος είναι ο μέγιστος χρόνος καθυστέρησης, ο οποίος εξαρτάται από την διαθέσιμη μνήμη RAM. Μπορούμε να υποθέσουμε πως μια τιμή γύρω στα 800 μs είναι υπεραρκετή για ένα απλό Digital Delay.² Οι απολύτως αναγκαίοι παράμετροι ελέγχου του συστήματος, οι οποίοι καθορίζονται από τον χρήστη μέσω δύο ποτενσιομέτρων, είναι ο καθορισμός του κέρδους της (αρνητικής) απόδοσης (Feedback gain) και ο χρόνος καθυστέρησης (Delay time).

Όσον αφορά την δομή του συστήματός μας, αυτό πρέπει να αποτελείται από δύο υποσυστήματα, ένα Αναλογικό και ένα Ψηφιακό. Η “καρδιά” του συστήματός είναι το ψηφιακό του υποσύστημα αφού εκεί θα εφαρμοστεί κάποιος αλγόριθμος επεξεργασίας των δειγμάτων. Χρειάζομαστε λοιπόν κάποιον μικροελεγκτή με την αναγκαία ποσότητα μνήμης RAM (εσωτερική ή εξωτερική) για να λειτουργήσει ο αλγόριθμος. Επίσης χρειαζόμαστε οπωσδήποτε δύο μετατροπείς ADC (Analog to Digital Converter) και DAC (Digital to Analog Converter), είτε ως ξεχωριστά chips, είτε ενσωματωμένους μέσα στον μικροελεγκτή. Επιπλέον είναι απαραίτητη η ύπαρξη κάποιας hardware πλατφόρμας προγραμματισμού για τον μικροελεγκτή που θα επλέξουμε, προκειμένου να τον προγραμματίσουμε χρησιμοποιώντας κάποια γλώσσα προγραμματισμού.

Σχετικά με το αναλογικό υποσύστημα, αυτό θα πρέπει οπωσδήποτε να περιέχει δύο βαθυπερατά (Low Pass) φίλτρα όμοια μεταξύ τους. Το πρώτο θα χρησιμοποιηθεί ως anti-aliasing φίλτρο προτού το αρχικό σήμα οδηγηθεί στον A-D μετατροπέα, ενώ μέσω του δεύτερου φίλτρου θα πρέπει να περάσει η έξοδος του D-A μετατροπέα, προκειμένου να εξαλειφθούν οι ανεπιθύμητες συχνότητες που προέκυψαν από την δειγματοληψία.

¹ Κάθε Φάση ανταποκρίνεται ένα σύνολο εργασιών που έχουν κάποια σχέση μεταξύ τους για την επίτευξη κάποιου επιμέρους στόχου. Τα πιο βασικά είναι η Σύλληψη Ιδέας, ο Σχεδιασμός, η Εκτέλεση, και η Παράδοση.

² Το γ. χρόνος της άμεσης απάντησης και η αρχικός χρόνος καθυστέρησης με το μ. χρόνο της διαθέσιμης μνήμης, σε συνδυασμό με την καθαρή απόδοση των τριών της μνήμης RAM τα τελευταία χρόνια, έχει οδηγήσει σε Delay Units με μέγιστο χρόνο πάνω από 2 sec. όπως περιγράφει με το μοντέλο DMI της Boss.

Από την προηγούμενη ανάλυση προκύπτει ότι υπάρχουν πολλές επιλογές που έπρεπε να γίνουν, οι οποίες όμως υπόκεινται στους περιορισμούς που αναφέρθηκαν και που είναι, αφενός μεν το να κρατηθεί χαμηλό το κόστος κτήσης των υλικών και αφετέρου η δυνατότητα να μπορεί να κατασκευαστεί το πρωτότυπο από εμάς, εύκολα και με ερασιτεχνικά εργαλεία. Η αναγκαιότητα για χαμηλό κόστος του project έκανε απαγορευτική την αγορά μιας πλατφόρμας ανάπτυξης βασισμένης σε κάποιον DSP επεξεργαστή η οποία θα είχε πάνω της όλα τα περιφερειακά, ADC, DAC, Audio In και Audio Out. Έτσι περιορίσαμε την επιλογή μας σε έναν μεμονωμένο φθινό μικροελεγκτή, ενώ την εντελώς απαραίτητη πλατφόρμα προγραμματισμού του, θα την κατασκευάσαμε εμείς αγοράζοντας τα αναγκαία εξαρτήματα.⁴

Ο δεύτερη σημαντική συνθήκη που αφορούσε την κατασκευή του πρωτοτύπου από εμάς, είχε ως συνέπεια τον περιορισμό της επιλογής μας για όλα τα υποψήφια ολοκληρωμένα (chips) του συστήματος, σε chips τύπου DIP (Dual In Line Package), η ύπαρξη των οποίων στις μέρες μας είναι όλο και πιο σπάνια.⁴

Φάση Σχεδιασμού (Planning)

Κατά την φάση αυτή έπρεπε να γίνει η επιλογή των ολοκληρωμένων που θα χρειάζονταν προκειμένου να γίνουν οι απαραίτητες παραγγελίες στους προμηθευτές. Όσον αφορά για την καρδιά του συστήματός μας που είναι ο μικροελεγκτής, καταλήξαμε στο καταπληκτικό *Proteiler chip* της εταιρίας *Parallax*, κυρίως γιατί είναι ο μοναδικός μικροελεγκτής που έχει ενσωματωμένη 32Kbytes στατική μνήμη RAM, και επιπλέον παράγεται σε συσκευασία 40pin DIP. Το κόστος του ολοκληρωμένου αυτού ήταν γύρω στα 15€, το οποίο είναι πραγματικά πολύ μικρό για τις εξαιρετικές του δυνατότητες, όπως θα δούμε σε επόμενο κεφάλαιο.

Να σημειώσουμε πως η εκτίμηση της αναγκαίας μνήμης RAM που απαιτείται για ένα τέτοιο σύστημα, έγινε στην φάση αυτή και υπολογίστηκε ως εξής:

Αν υποθέσουμε ότι θα χρησιμοποιήσουμε συχνότητα δειγματοληψίας $f_s = 20 \text{ KHz}$ και κάθε δείγμα αποθηκεύεται σε ένα Ram cell μεγέθους $k = 16 \text{ bits}$, τότε προκειμένου να αποθηκεύσουμε ηχητικό σήμα διάρκειας $t = 800 \text{ ms} = 0.8 \text{ sec}$ χρειάζομαστε ποσότητα μνήμης ίσης προς:

$$L = f_s k t = 256000 \text{ bits} \approx 32 \text{ Kbytes}$$

Σχετικά με τους δύο μετατροπείς (A-D, D-A) αρχικά υπήρξε μεγάλο πρόβλημα, αφού δυστυχώς το Proteiler chip δεν έχει ενσωματωμένο ADC ή DAC, και γενικά δεν υπάρχουν σε συσκευασία DIP μετατροπείς κατάλληλοι για ηχητικό σήμα. Στο σημείο αυτό λοιπόν, επιλέξαμε τους DIP μετατροπείς MC3202, MCP1821 της εταιρίας *Microchip Technology*, για ADC και DAC αντίστοιχα. Η εταιρία *Microchip Technology* είχε την ευγενή καλοσύνη να μας προμηθεύσει δωρεάν όσα δείγματα της ζητήσαμε, διευκολύνοντας σε μεγάλο βαθμό την γρήγορη ανάπτυξη του πρωτοτύπου. Οι προδιαγραφές τους είναι ότι λειτουργούν με μέγιστη ανάλυση τα 12 bits ενώ η μέγιστη συχνότητα δειγματοληψίας του ADC είναι τα 100 *ksamples/sec*. Και τα δύο chips χρησιμοποιούν το πρωτόκολλο επικοινωνίας SPI. Όπως θα δούμε στο τελευταίο κεφάλαιο, εκ του ηχητικού αποτελέσματος τα δύο αυτά ολοκληρωμένα δεν έχουν τίποτα να ζητήσουν από εξειδικευμένους σε σήματα ήχου μετατροπείς.

Με το αναλογικό μέρος η κατάσταση ήταν πιο εύκολη αφού ήταν δεδομένη η χρήση τελεστικών ενισχυτών (op-amps) για την δημιουργία των δύο φίλτρων, όπου η προμήθεια τους είναι γενικά εύκολη και με πολύ μικρό κόστος. Έτσι διαλέξαμε τον διπλό τελεστικό 5532, ο οποίος είναι δοκιμασμένος για πολλά χρόνια σε ηχητικές εφαρμογές με εξαιρετικά αποτελέσματα. Ο μοναδικός περιορισμός που θα προέκυπτε από την επιλογή του 5532, ήταν η χρήση κατάλληλου τροφοδοτικού που θα μας έδινε διπολική τάση (bipolar supply) +/- 10 Volts, αφού ο τελεστικός αυτός δεν μπορεί να λειτουργήσει ικανοποιητικά με μονοπολική τάση (single supply) στα 5 Volts.

Εκτός από τα δύο φίλτρα, είναι επίσης αναγκαία η ύπαρξη ενός τελεστικού ως input buffer στην είσοδο και ενός τελεστικού ως εξαιλενγητή στην έξοδο, προκειμένου αφενός μεν να μπορεί συνδεθεί χωρίς κατώλειες στην είσοδο του επόμενου πηγή μεγάλης σύνθετης αντίστασης (όπως είναι η ηλεκτρική κιθάρα) και αφετέρου το πλάτος των σημάτων εισόδου και εξόδου να είναι περίπου το ίδιο. Τέλος να σημειώσουμε πως επειδή το σύστημά μας προορίζεται για σύνδεση με ηλεκτρική κιθάρα, το εύρος συχνοτήτων του ουσιαστικά δεν ξεπερνά τα 5 KHz.

Μετά από όλα αυτά έχουμε τον πίνακα 4.1, όπου συνοψίζουμε τις αρχικές Προδιαγραφές - Απαιτήσεις του συστήματος.

⁴Μην πάτε να αντιμετωπίσετε με κάποιο τρόπο, καθιστώντας άμεσο το σύστημα ή γρήγορο αδύνατο.

⁴Όπως είδαμε οι περισσότερες εταιρίες που φτιάχνουν ολοκληρωμένα υλικά, στα κυρίως για λόγους κόστους δεν χρησιμοποιούν τέτοια packages για τα κοινωτικά chips που παράγουν, με αποτέλεσμα να είναι σχεδόν αδύνατη η κατασκευή πρωτοτύπων με σωστό τρόπο.

Αρχικές Προδιαγραφές - Απαιτήσεις	
Μέγιστος Χρόνος Καθυστέρησης	800 ns
Συχνότητα δειγματοληψίας	20 KHz
Ανάλυση	12 bits
Μικροελεγκτής	Propeller chip
A-D Μετατροπέας	MCP3202 (Successive Approximation, 12 bit)
D-A Μετατροπέας	MCP1821 (Resistive String, 12 bit)
Μνήμη	32Kbytes static Ram ενσωματωμένη στο Propeller chip
Παράμετροι Ελέγχου	Delay time, feedback gain
Τάση τροφοδοσίας	+/- 1V Volts
Απόκριση Συχνότητων	20 Hz – 5 KHz

Πίνακας 1.1: Αρχικές απαιτήσεις-προδιαγραφές του συστήματος

Φάση “Κατασκευής” (Production).

Στην φάση αυτή αναπτύξαμε σταδιακά τα διάφορα υφιστάμια που απαιτούντο το σύστημα, κατά την διάρκεια της οποίας είχαμε την τμηματική παραλαβή των υλικών. Αρχικά κατασκευάσαμε μια πλατφόρμα προγραμματισμού για το Propeller chip, μέσω των διαθέσιμων σχεδίων που έχει η εταιρία Parallax στο site της ([SerialToPropeller.pdf](#)), χρησιμοποιώντας την σειριακή θύρα για επικοινωνία με ένα απλό PC (και όχι την πιο πολύπλοκη θύρα usb). Να σημειώσουμε πως κατά την φάση αυτή αποφασίσαμε να “σπάσουμε” το ψηφιακό υποσύστημα μας σε δύο φυσικά μέρη, όπου το πρώτο είναι η πλατφόρμα προγραμματισμού, ενώ το δεύτερο είναι ένα ξεχωριστό daughter board που περιλαμβάνει τους δύο A-D και D-A μετατροπείς, μαζί με τα δύο ποτενσιόμετρα μέσω των οποίων ο χρήστης καθορίζει τις παραμέτρους Feedback gain και Delay time.

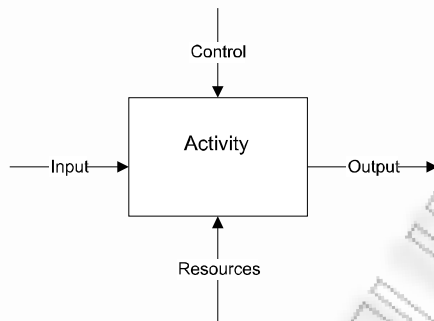
Αφού είχαμε στην διάθεση μας την πλατφόρμα προγραμματισμού άρχισε η σταδιακή (και χρονοβόρα) ανάπτυξη του firmware χρησιμοποιώντας τις Texas Instruments γλώσσες προγραμματισμού: [Propbasic](#) και [Assembly](#), εφαρμόζοντας μια προσέγγιση “από κάτω προς τα πάνω” (down to top design). Αρχικά λοιπόν καθορίσαμε τις προγραμματιστικές μονάδες από τις οποίες πρέπει να συγκροτείται ένα τέτοιο πρόγραμμα. Αυτές αφορούν, την επικοινωνία με το ADC chip, την επικοινωνία με το DAC chip, τον αλγόριθμο του Delay Effect, και τέλος την αλληλεπίδραση με τα δύο ποτενσιόμετρα που καθορίζουν τις παραμέτρους ελέγχου. Κάθε ένα από αυτά τα “υφιστάμια” του προγράμματος, σχεδιάστηκε, αναπτύχθηκε και ελέγχθηκε ανεξάρτητα από τα υφιστάμια, προκειμένου να είμαστε σίγουροι για την ορθότητα και την αξιοπιστία κάθε τμήματος, όταν θα έφτανε η ώρα της τελικής συναρμολόγησης και σύνθεσής τους.

Τέλος όσον αφορά την κατασκευή του αναλογικού υποσυστήματος δεν υπήρξαν σημαντικές αναθεωρήσεις της αρχικής σχεδίασης. Το μόνο στοιχείο που προστέθηκε στο τέλος, ήταν ένας διακόπτης Bypass, ο οποίος επιλέγει ως έξοδο του κυκλώματος, είτε το ανεπεξέργαστο αρχικό σήμα (dry signal), είτε το ψηφιακά επεξεργασμένο καθυστερημένο σήμα (wet signal), προκειμένου να μπορούμε να εκτιμήσουμε την επίδραση του εφέ στο σήμα εισόδου.

Κεφάλαιο 5

Περιγραφή του συστήματος Digital Delay

5.1 Διαγραμματική μέθοδος ανάλυσης συστημάτων IDEF0



Σχήμα 5.1: Activity box: Το δομικό στοιχείο της μεθόδου IDEF0.

Προκειμένου να περιγράψουμε και να αναλύσουμε το σύστημα *Digital Delay* που αναπτύξαμε, θα χρησιμοποιήσουμε την διαγραμματική μέθοδο IDEF0 (Integrated computer aided manufacture Definition 0), οπότε κρινεται σκόπιμη μια σύντομη περιγραφή της μεθόδου αυτής.

Τα διαγράμματα IDEF0 ανήκουν στις μεθόδους δομημένης ανάλυσης συστημάτων και έχουν πολλές ομοιότητες με τα συνηθή διαγράμματα ροής δεδομένων (Data Flow Diagrams), με περισσότερες όμως δυνατότητες.

Το δομικό στοιχείο των διαγραμμάτων αυτών είναι το "κουτί λειτουργίας" (activity box) το οποίο περιγράφει μια λειτουργία του συστήματος, η οποία μπορεί να είναι μια διαδικασία επεξεργασίας ή μετασχηματισμού μιας πληροφορίας. Σε κάθε κουτί λειτουργίας φαίνονται τα αντικείμενα (πληροφορίες, υλικά) που χρησιμοποιεί ως εισόδους, τα αντικείμενα που αποτελούν τις εξόδους της λειτουργίας, οι μεταβλητές ελέγχου και οι εξωτερικοί περιορισμοί στους οποίους υπόκειται η λειτουργία και οι πόροι-μηχανισμοί (πχ Μηχανές, Λογισμικό) που αυτή χρησιμοποιεί. (βλ. σχήμα 5.1)

Η μέθοδος IDEF0 οργανώνει τις λειτουργίες του συστήματος με ιεραρχικό τρόπο. Κάθε διάγραμμα είναι είτε συγκεντρωτικό (δηλαδή διάγραμμα-γονέας) είτε λεπτομερές (διάγραμμα-παιδί). Κάθε λειτουργία αν κριθεί απαραίτητο αναλύεται στις υπολειτουργίες από τις οποίες αποτελείται. (μέγιστο πλάτος 6), οι οποίες και θα αποτελέσουν ένα νέο διάγραμμα-παιδί χαμηλότερου επιπέδου (layer). Σε κάθε τέτοια αποσύνθεση (decomposition) το διάγραμμα παιδί κληρονομεί τις εισόδους και εξόδους του διαγράμματος γονέα. Η ανάλυση σε χαμηλότερα επίπεδα συνεχίζεται μέχρι τον επιθυμητό βαθμό λεπτομέρειας στην παράσταση του συστήματος. Πρέπει να σημειώσουμε ότι, οι λειτουργίες που απαρτίζουν ένα διάγραμμα συνδέονται μεταξύ τους κυρίως μέσω των εισόδων-εξόδων τους αλλά δεν αποκλείεται για παράδειγμα η έξοδος μιας δραστηριότητας να είναι περιεχόμενος μιας άλλης. Επίσης σε κάθε διάγραμμα οι δραστηριότητες δεν τοποθετούνται τυχαία αλλά υπέρχει διάταξη (staircase pattern) η οποία καθορίζεται από τον τρόπο που αυτές σχετίζονται (πχ χρονική σειρά) [2].

5.2 Λειτουργική περιγραφή του ψηφιακού συστήματος Delay-Echo

Μετά από σύντομη πρόλογη περιγραφή της διαγραμματικής μεθόδου IDEF0, είμαστε σε θέση να περιγράψουμε το τελικό σύστημά μας με την βοήθεια του συγκεντρωτικού διαγράμματος του σχήματος 5.2. Όπως βλέπουμε το σύστημα (που συμβολίζεται με τον κόμβο A0) παίρνει ως είσοδο ένα αναλογικό ηχητικό σήμα I_1 : *Audio In* το οποίο μετασχηματίζει σε ένα επίσης αναλογικό ηχητικό σήμα O_1 : *Audio Out*. Οι παράμετροι ελέγχου του συστήματος που καθορίζονται από την χρήση είναι ο χρόνος καθυστέρησης, C_1 : *Delay time*, το κέρδος της ανάδρασης, C_2 : *Feedback gain* και ο δ.α.κ.α. της ενεργοποίησης-απενεργοποίησης, C_3 : *By Pass*. Οι πόροι που χρησιμοποιεί το σύστημα για να επιτευχθεί η λειτουργία της καθυστέρησης του σήματος

εισόδου είναι η τάση τροφοδοσίας, R_1 : *Higher supply* και το λογισμικό, R_2 : *Software* (που μπορεί να είναι αποθηκευμένο σε μνήμη *Erprom*) το οποίο εκτελείται από την συσκευή.

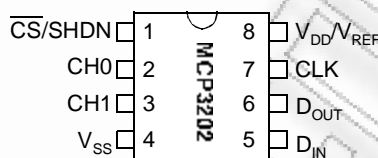
Στο σχήμα 5.3 βλέπουμε ότι το σύστημά μας συνίσταται από δύο κύρια υποσυστήματα, το Αναλογικό υποσύστημα, *Analog Subsystem* (κώδικας A1) και το Ψηφιακό υποσύστημα, *Digital Subsystem* (κώδικας A2). Επίσης από το διάγραμμα αυτό μπορούμε να δούμε και το πως κατανομούνται τα “αντικείμενα” I_1, O_1, R_i, C_i στα δύο αυτά υποσυστήματα. Έτσι οι παράμετροι ελέγχου *Delay time* και *Feedback gain*, έχουν να κάνουν με το ψηφιακό υποσύστημα, ενώ ο διακόπτης *By Pass* με το αναλογικό υποσύστημα. Κάτι άλλο που προκύπτει από το διάγραμμα αυτό, είναι ότι το αναλογικό υποσύστημα είναι υπεύθυνο για την παραγωγή της τάσεως των 5 Volts που απαιτείται για την λειτουργία του ψηφιακού υποσυστήματος. Τέλος στα σχήματα 5.4 και 5.5 μπορούμε να δούμε αναλυτικά τις λειτουργίες που εκτελούνται στα δύο υποσυστήματα, τις οποίες θα αναλύσουμε στις δύο επόμενες ενότητες.

Αν θέλαμε να περιγράψουμε τα στάδια από τα οποία περνάει το αρχικό σήμα εισόδου I_1 μέχρι να μετασχηματιστεί στο σήμα εξόδου O_1 μπορούμε να πούμε συνοπτικά τα εξής: Το σήμα εισόδου αφού περάσει από το *Input Buffer* οδηγείται σε ένα βαθυπερατό φίλτρο (*ADC filter*) και κατόπιν στον μετατροπέα *ADC*. Στην συνέχεια το ψηφιακό πια σήμα υφίσταται ψηφιακή επεξεργασία από το *Processor board* μέσω του λογισμικού που έχουμε γράψει με παραμέτρους τα *Delay time*, *feedback gain* και οδηγείται στον μετατροπέα *DAC* όπου μετατρέπεται σε αναλογικό σήμα. Το σήμα αυτό (*DAC Output*) περνάει από ένα βαθυπερατό φίλτρο (*DAC Filter*) και στην συνέχεια αφού μειωθεί το πλάτος του σε επίπεδα ανάλογα με το σήμα εισόδου (*Attenuator*), δρομολογείται στην έξοδο μέσω ενός αναλογικού διακόπτη (*By Pass Switch*). Ο διακόπτης αυτός είναι τύπου *SMD* (*Single Pole Double Throw*), και λειτουργεί ως επιλογήας. Στην μία θέση του επιλέγει το ανεπεξεργαστο σήμα (*dry signal*) προερχόμενο από το *Input Buffer*, ενώ στην άλλη θέση του επιλέγει το επεξεργασμένο σήμα (*wet signal*) ως σήμα εξόδου.

5.3 Ανάλυση του ψηφιακού υποσυστήματος

Στα σχήματα 5.5 και 5.19, μπορούμε να δούμε αναλυτικά τα διάφορα τμήματα από τα οποία αποτελείται το ψηφιακό υποσύστημα καθώς και την αλληλεπίδραση τους μέσω των διαφόρων σημάτων εισόδου-εξόδου και των σημάτων ελέγχου. Έτσι βλέπουμε πως έχουμε έναν *A-D* μετατροπέα (*MCP3202*), τον μικροελεγκτή (*Processor board*), τον *D-A* μετατροπέα (*MCP1821*) και τα δύο ποτενσιόμετρα που ελέγχουν τις παραμέτρους *Feedback gain* και *Delay time*.

5.3.1 A-D μετατροπέας (MCP3202)

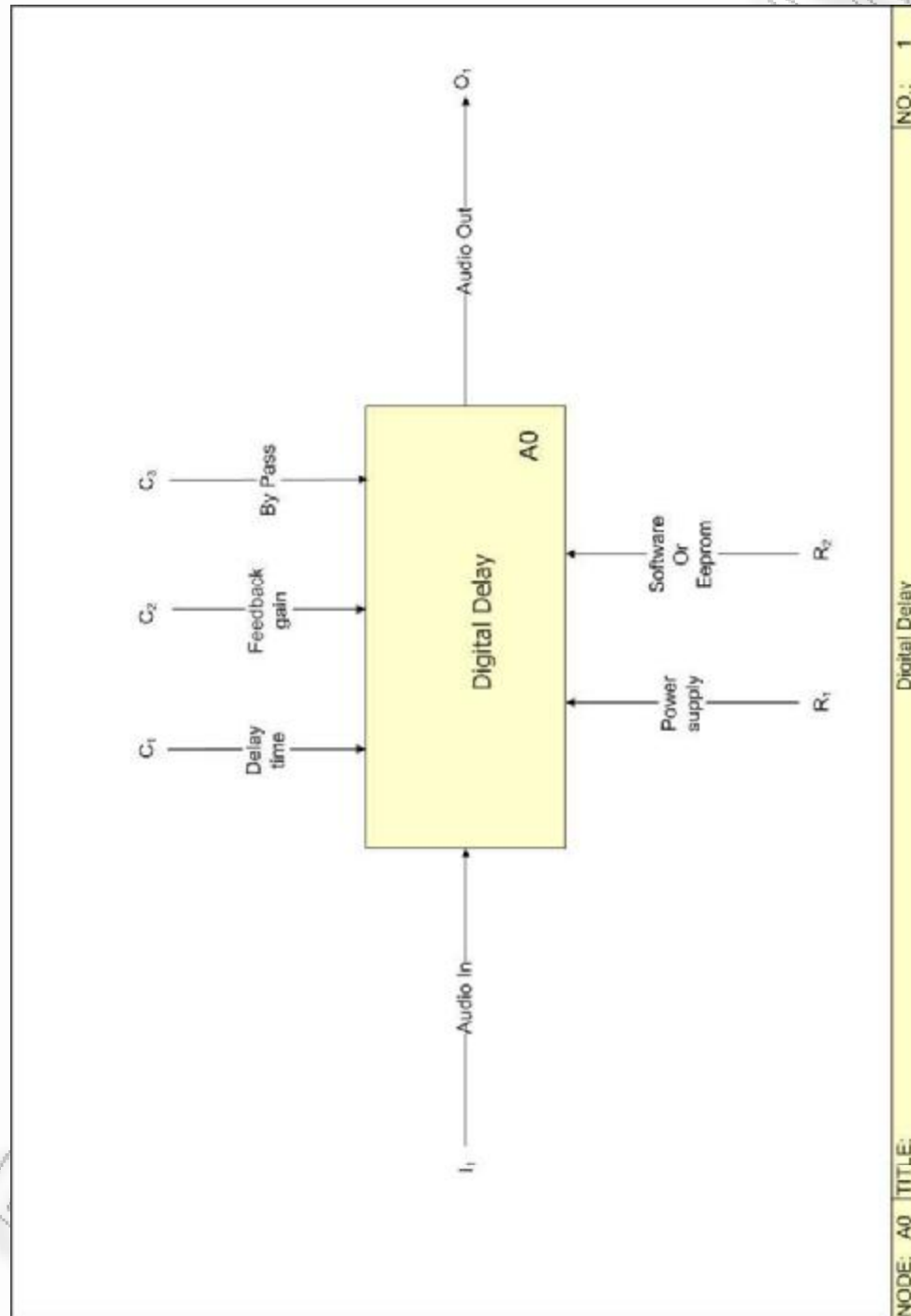


Σχήμα 5.6: MCP3202 pinout

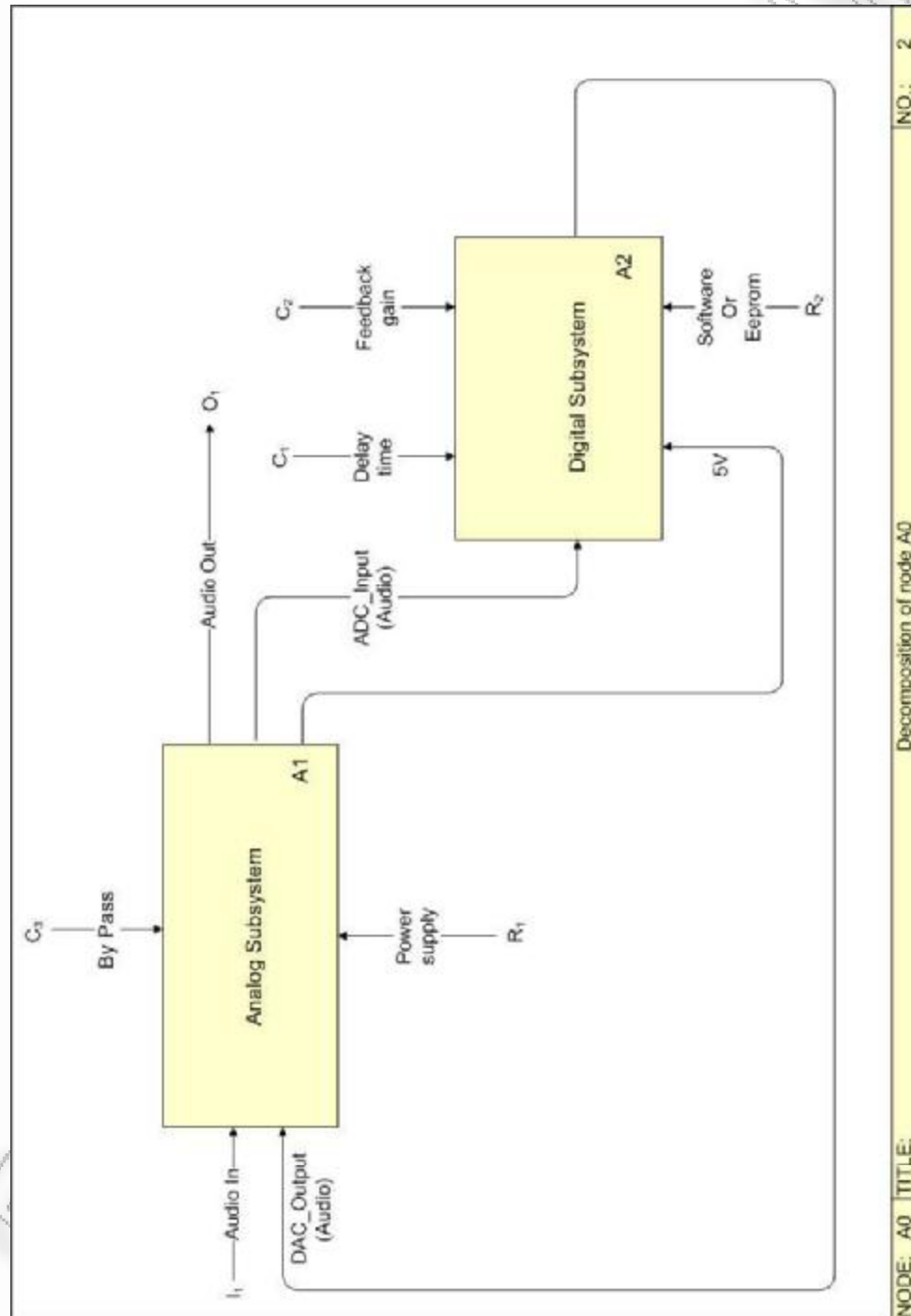
Όπως έχουμε ήδη αναφέρει ο *A-D* μετατροπέας που χρησιμοποιήσαμε, είναι ο *MCP3202* της εταιρίας *Microchip*. Ο μετατροπέας αυτός χρησιμοποιεί την μέθοδο των διαδοχικών προσεγγίσεων (*successive approximation*) προκειμένου να πετύχει την μετατροπή, όπως φαίνεται στο λειτουργικό του διάγραμμα (σχήμα 5.7 [6]). Κατά την μέθοδο αυτή, η αναλογική τάση V_{in} που θέλουμε να μετατρέψουμε εφαρμόζεται στην πρώτη είσοδο ενός συγκριτή (*comparator*), ενώ μια χρονικά μεταβαλλόμενη τάση αναφοράς V_{ref} προερχόμενη από ένα *DAC*, εφαρμόζεται στην άλλη είσοδο του συγκριτή. Η τάση V_{ref} αρχικά παίρνει την τιμή $V_{max}/2$ και εφαρμόζεται ο γνωστός αλγόριθμος της διαδοχικής αναίτησης περιορίζοντας κάθε φορά το εύρος του διαστήματος αναίτησης στο μισό του προηγούμενου, εξαρτώμενο από το αποτέλεσμα κάθε σύγκρισης, με αποτέλεσμα να έχουμε μια πολύ γρήγορη μετατροπή. Το αποτέλεσμα κάθε μίας σύγκρισης μας δίνει και αυτό ένα bit ξεκινώντας από το *MSB* και καταλήγοντας στο *LSB*. Επειδή η μετατροπή γίνεται bit προς bit είναι πιο εύκολη η κατασκευή ολοκληρωμένων που μας δίνουν το αποτέλεσμα της μετατροπής σειριακά όπως συμβαίνει με το *MCP3202*. Να σημειώσουμε πως προκειμένου η μετατροπή να γίνει σωστά, είναι απολύτως απαραίτητο η τάση V_{in} να παραμείνει σταθερή καθ' όλη την διάρκεια της μετατροπής, με αποτέλεσμα να χρησιμοποιείται ένα κύκλωμα “*sample and hold*” προκειμένου αυτό να γίνει εφικτό. Στον πίνακα 5.1 μπορούμε να δούμε μερικά τεχνικά χαρακτηριστικά του *MCP3202* για τάση λειτουργίας τα 5Volts.

Στο σχήμα 5.6 [6], βλέπουμε τους 8 ακροδέκτες του ολοκληρωμένου. Παρατηρούμε πως δεν υπάρχει ειδικός ακροδέκτης για την τάση αναφοράς, αφού η τάση τροφοδοσίας είναι και η τάση αναφοράς V_{ref} . Ττσι επειδή εμείς θα τροφοδοτήσουμε το ολοκληρωμένο με τάση 5 Volts αυτό σημαίνει πως και η $V_{ref} = 5 Volts$. Επίσης να σημειώσουμε πως στην κατασκευή μας χρησιμοποιήσαμε και τις δύο εισόδους *CH0*, *CH1* (ακροδέκτες 2 και 3), θέτοντας το ολοκληρωμένο σε “*pseudo-differential mode*” για να έχουμε λιγότερο θόρυβο, σύμφωνα με το φύλλο οδηγιών [6].

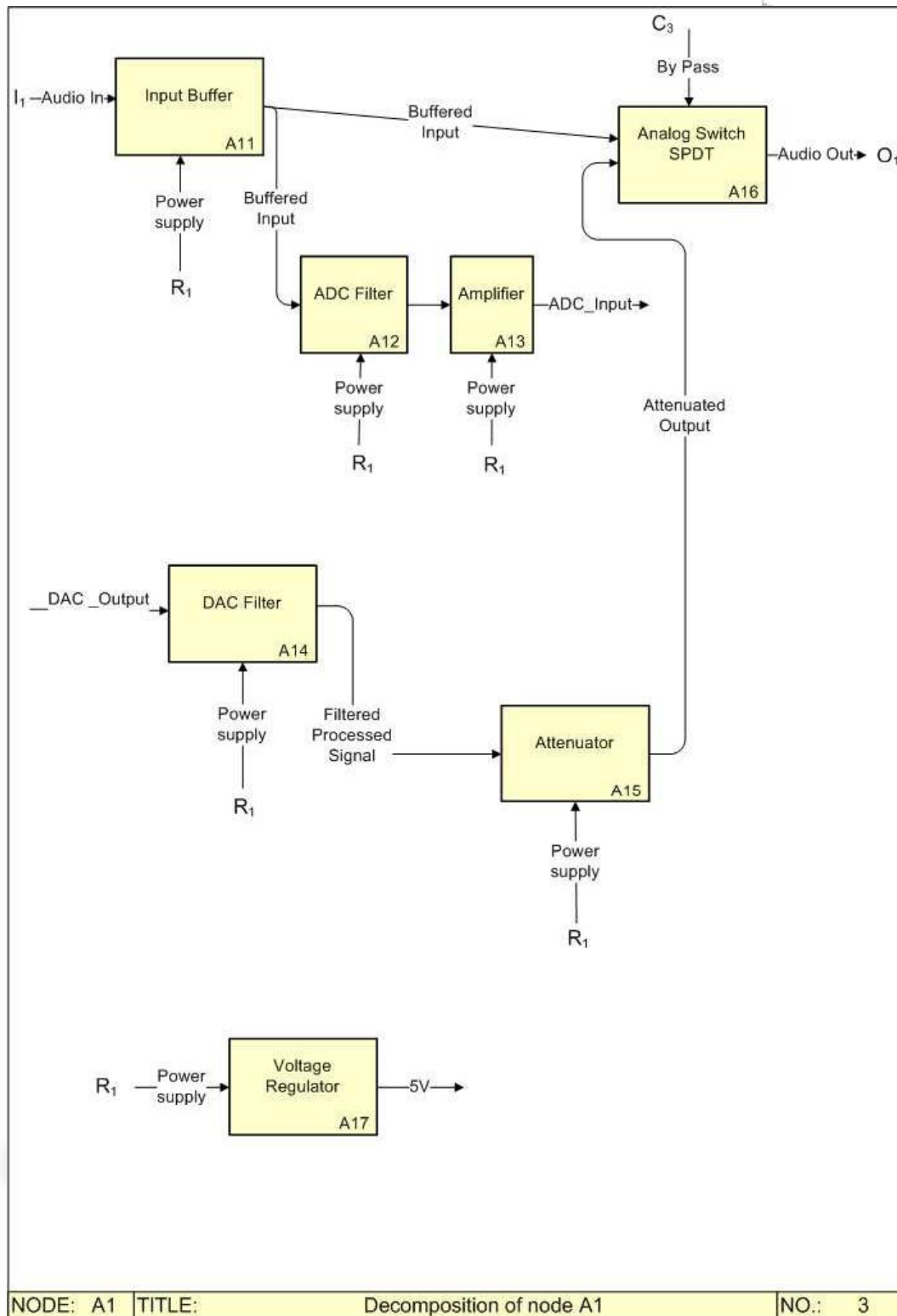
Στο σχήμα 5.6 [6], βλέπουμε τους 8 ακροδέκτες του ολοκληρωμένου. Παρατηρούμε πως δεν υπάρχει ειδικός ακροδέκτης για την τάση αναφοράς, αφού η τάση τροφοδοσίας είναι και η τάση αναφοράς V_{ref} . Ττσι επειδή εμείς θα τροφοδοτήσουμε το ολοκληρωμένο με τάση 5 Volts αυτό σημαίνει πως και η $V_{ref} = 5 Volts$. Επίσης να σημειώσουμε πως στην κατασκευή μας χρησιμοποιήσαμε και τις δύο εισόδους *CH0*, *CH1* (ακροδέκτες 2 και 3), θέτοντας το ολοκληρωμένο σε “*pseudo-differential mode*” για να έχουμε λιγότερο θόρυβο, σύμφωνα με το φύλλο οδηγιών [6].



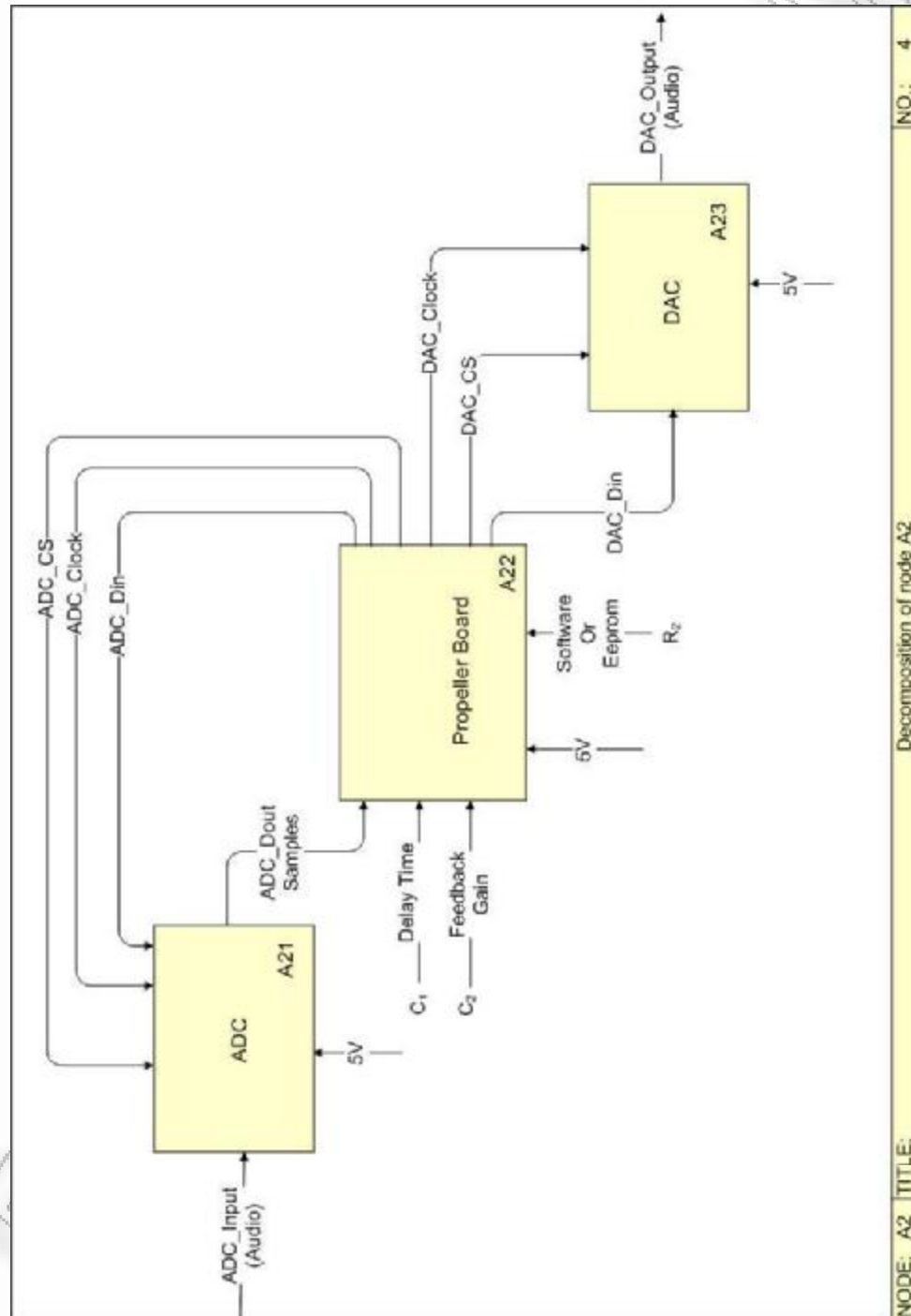
Σχήμα 5.2: Άποψη του υψηλότερου επιπέδου του συστήματος: Κόμβος A0



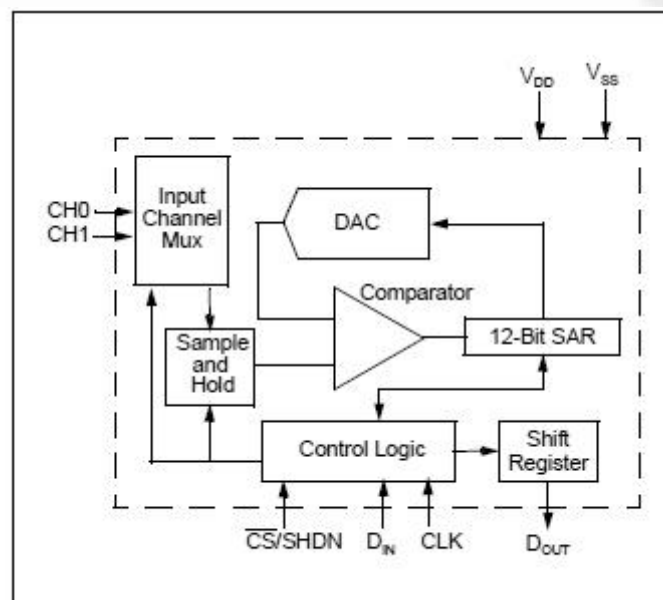
Σχήμα 5.3: Αποσύνθεση κόμβου A0



Σχήμα 5.4: Αποσύνθεση του Αναλογικού υποσυστήματος (Κόμβος A1)



Σχήμα 5.5: Αποσύνθεση του Ψηφιακού υποσυστήματος (Κόμβος A2)



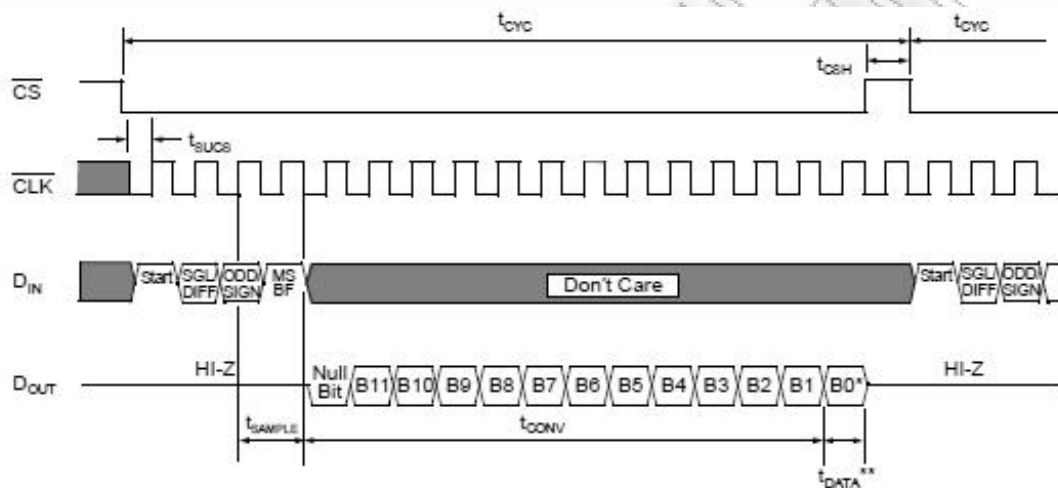
Σχήμα 3.7: Δε. τουργικό διάγραμμα του MCP3202

Τεχνικά Χαρακτηριστικά του MCP3202 για $V_s=5$ Volts	
Όνομαστική ανάλυση	12 bits
Μέγιστη συχνότητα δειγματοληψίας	100 ksps
Μέγιστη συχνότητα ρολογιού	1.8 MHz
Περίοδος Δειγματοληψίας	18 κύκλοι ρολογιού
Αριθμός καναλιών εισόδου	2 (Single-Ended Mode) ή 1 (Pseudo-Differential Mode)
Πρωτόκολλο επικοινωνίας	I ² C

Πίνακας 3.1: Μερικά τεχνικά χαρακτηριστικά του MCP3202

	Config Bits		Channel Selection		GND
	Sgl/Diff	Odd/sign	0	1	
Single Ended Mode	1	0	+	-	-
	1	1	-	+	-
Pseudo-Differential Mode	0	0	IN+	IN-	-
	0	1	IN-	IN+	-

Σχήμα 5.8: Configuration bits για το MCP3202.



Σχήμα 5.9: Επικοινωνία με το MCP3202. Κάθε ψηφιοποιημένο δείγμα το λαμβάνουμε (μια φορά) σειριακά σε μορφή "MSB first".

Η επικοινωνία με το chip πραγματοποιείται σύμφωνα με το σειριακό πρωτόκολλο SPI, χρησιμοποιώντας τους ακροδέκτες D_{in} , D_{out} , CLK . Μπορούμε να επιλέξουμε το chip, προκειμένου να επικοινωνήσουμε μαζί του, μέσω του (active low) ακροδέκτη, \overline{CS} (chip select). Έτσι η επικοινωνία αρχικοποιείται με το να θέσουμε τον ακροδέκτη \overline{CS} σε χαμηλό δυναμικό (εφόσον πριν ήταν σε υψηλό), όπως βλέπουμε στη σχήμα 5.9 [6]. Κατόπιν πρέπει να καθορίσουμε τον τρόπο λειτουργίας του chip στέλνοντας, σειριακά στην γραμμή D_{in} , τέσσερα συνεχόμενα bits. Το πρώτο high bit που θα στείλουμε είναι το start bit. Ακολουθούν δύο bits οι τιμές των οποίων καθορίζουν εάν θα χρησιμοποιήσουμε το single ended ή το pseudo-differential mode, σύμφωνα με τον πίνακα 5.8. Τέλος το είδος του τέταρτου bit θα καθορίσει το αν το ολοκληρωμένο θα μας στείλει το αποτέλεσμα της μετατροπής και με την μορφή "LSB (Least Significant Bit) first", αφού έχει ολοκληρωθεί η by default εκπομπή κατά την οποία θα μας στείλει πρώτα το MSB (Most Significant Bit). Έτσι αν το τέταρτο bit είναι 1 τότε έχουμε την by default συμπεριφορά, ενώ αν είναι 0 θα έχουμε και επιπλέον εκπομπή του αποτελέσματος σε μορφή "LSB first".

Μετά στον χρόνο που εμείς στέλνουμε τα 4 configuration bits, το ολοκληρωμένο έχει ολοκληρώσει την μετατροπή (η οποία διαρκεί 1.5 κύκλο ρολογιού) και έτσι αφού αυτό λάβει και το τέταρτο bit, θα μας στείλει στην γραμμή D_{out} ένα low bit (null bit) για να μας πληροφορήσει ότι θα ακολουθήσει η εκπομπή των 12 bits της μετατροπής, όπου το πρώτο από αυτά θα είναι Most Significant Bit (MSB first). Στο σημείο αυτό πρέπει να σημειώσουμε πως ο μετατροπέας λαμβάνει κάθε ένα από τα 4 configuration bits στην γραμμή D_{in} , κατά την μετάβαση της γραμμής CLK από χαμηλό σε υψηλό δυναμικό, ενώ κάθε ένα εκ των 13 bits (12 bits+null bit) που μας στέλνει, γίνεται διαθέσιμο στην γραμμή D_{out} κατά την μετάβαση της γραμμής CLK από υψηλό σε χαμηλό δυναμικό. Τέλος αφού λάβουμε και το 12^ο bit της μετατροπής, απενεργοποιούμε το chip θέτοντας τον ακροδέκτη \overline{CS} σε υψηλό δυναμικό και επαναλαμβάνουμε την διαδικασία για την λήψη του επόμενου δείγματος.

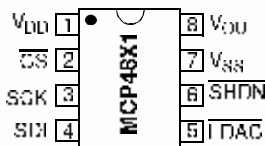
Μετά από όλα αυτά είναι φανερό πως στην περίπτωση μας, όπου θέλουμε ο μετατροπέας να λειτουργήσει

σε pseudo-differential mode, θα πρέπει να στείλουμε μέσω της γραμμής Din την ακολουθία 1001. Όπως θα δούμε στην επενότητα που θα εξετάσουμε το ProPeller chip, η συχνότητα του ρολογιού που θα τροφοδοτήσουμε το MCP3202 στον ακροδέκτη CLK, είναι $f_{CLK} = 625 \text{ KHz}$. Επειδή η περίοδος δειγματοληψίας, t_{OYC} , διαρκεί 18 κύκλους ρολογιού (βλ. σχήμα 5.9), η μέγιστη επιτρεπόμενη συχνότητα δειγματοληψίας που μπορούμε να επιτύχουμε θα είναι $f_{s(max)} = 625/18 \approx 35 \text{ KHz}$. Τέλος να σημειώσουμε πως η ψηφιακή τιμή των 12 bits που θα προκύψει από τον μετατροπέα για αναλογική είσοδο V_{in} , δίνεται από τον τύπο:

$$\text{DigitalOutputCode} = \frac{4096 \times V_{in}}{V_{ref}}$$

Επίσης η διακριτική ικανότητα του μετατροπέα θα είναι ίση προς $V_{ref}/2^n = 5 \text{ Volts}/4096 = 1.22 \text{ mV}$. Το τελευταίο σημαίνει πως η μικρότερη μεταβολή της τάσεως εισόδου που μπορεί να γίνει αντιληπτή από τον μετατροπέα είναι τα 1.22 mV.

5.3.2 D-A μετατροπέας (MCP4821)



Σχήμα 5.10: MCP4821 pinout

Ο μετατροπέας ψηφιακού σήματος σε αναλογικό που χρησιμοποιήσαμε για το σύστημά μας είναι ο MCP4821 της εταιρίας Microchip. Ο μετατροπέας αυτός έχει ονομαστική ανάλυση τα 12 bits και δημιουργεί ο ίδιος εισαγωγικά την απαραίτητη τάση αναφοράς V_{ref} στα 2.048 Volts. Όπως μπορούμε να δούμε από το λειτουργικό του διάγραμμα (σχήμα 5.11 [7]), στην έξοδό του υπάρχει τελεστικός ενισχυτής με προγραμματιζόμενο κέρδος, ο οποίος μπορεί να οδηγήσει, τόσο ωμικά, όσο και χωρητικά φορτία χωρίς πρόβλημα. Στο σημείο αυτό πρέπει να σημειώσουμε πως στο φύλλο οδηγιών του μετατροπέα (βλ. αναφορά [7]) δεν αναφέρεται χρήση του ολοκληρωμένου αυτού για ηχητικά σήματα. Παρ' όλα αυτά τα ηλε-

κτρικά χαρακτηριστικά του ενσωματωμένου τελεστικού, όπως είναι το ανεκτό Skew Rate=0.55 V/μs² και η δυνατότητα του να μπορεί να αποδίδει τάσεις σε όλο σχεδόν το εύρος της τάσεως τροφοδοσίας, τον καθιστά κατάλληλο για σήματα που εκτείνονται συχνотικά μέχρι τα 5 KHz.

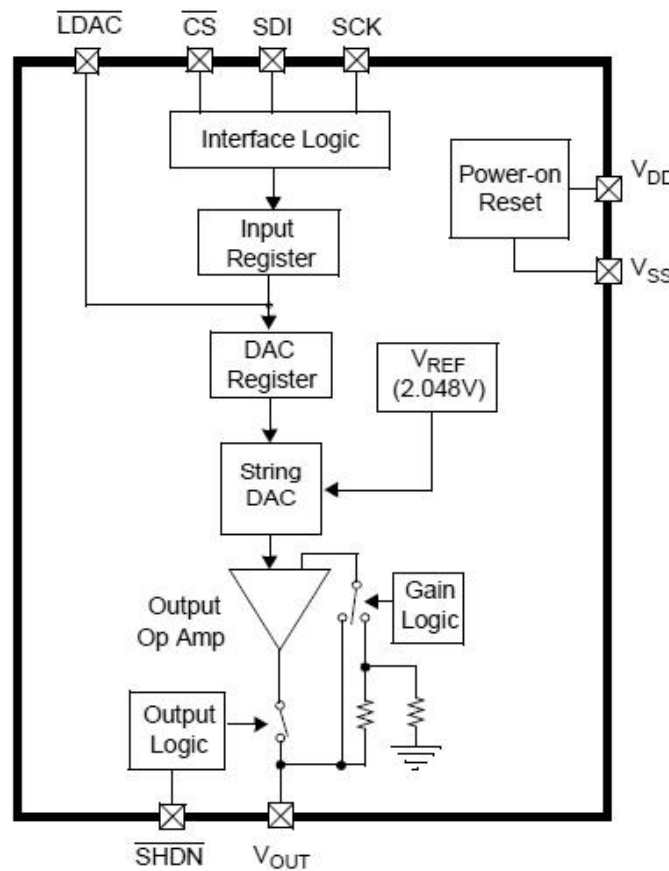
Στο σχήμα 5.10 [7], μπορούμε να δούμε τους 8 ακροδέκτες του ολοκληρωμένου. Έτσι στον ακροδέκτη 1 (VDD) εφαρμόζουμε την τάση τροφοδοσίας (2.7 V έως 3.3 V), ενώ ο ακροδέκτης 7 (VSS) πρέπει να γειωθεί. Η αναλογική έξοδος του μετατροπέα βρίσκεται στον ακροδέκτη 8, Vout. Όπως έχουμε αναφέρει η επικοινωνία με την συσκευή γίνεται με βάση τα πρωτόκολλο SPI. Εδώ να σημειώσουμε πως επειδή δεν υπάρχει ανάγκη για εκπομπή ψηφιακών δεδομένων από την μεριά του chip, δεν υπάρχει γραμμή Dout κι έτσι έχουμε μόνο δυο γραμμές αφιερωμένες για την επικοινωνία, την γραμμή SCK (pin 3) όπου μπορούμε να συνδέσουμε ένα ρολόι με μέγιστη συχνότητα τα 20 MHz, και την γραμμή SDI (pin 4) όπου μπορούμε να στείλουμε στον μετατροπέα ψηφιακά δεδομένα.

Προκειμένου να επικοινωνήσουμε με το ολοκληρωμένο πρέπει καταρχήν να το ενεργοποιήσουμε, κρατώντας τον ακροδέκτη 2 (\overline{CS}) σε χαμηλό δυναμικό. Κατόπιν, πρέπει να στείλουμε σειριακά μέσω της γραμμής SDI (pin4), τα απαραίτητα ψηφιακά δεδομένα τα οποία συνίστανται από μια "λέξη" (16 bits). Τα πρώτα 4 από αυτά είναι τα configuration bits, ενώ τα υπόλοιπα 12 bits (data bits) είναι η ψηφιακή τιμή την οποία θέλουμε να μας βγάλει μέσω αυτής του chip. Η μεταφορά προς το chip, για κάθε bit που κοι στέλνουμε μέσω της γραμμής SDI, πραγματοποιείται κατά την μετάβαση του παλμού του ρολογιού (στην γραμμή SCK) από χαμηλό σε υψηλό δυναμικό όπως φαίνεται στο σχήμα 5.12 [7]. Τέλος όταν στείλουμε και το τελευταίο 16^ο bit, θέτουμε τον ακροδέκτη 2 (\overline{CS}) σε υψηλό δυναμικό με αποτέλεσμα η "λέξη" που στείλαμε να μεταφερθεί στον Input register. Θέτοντας τώρα τον ακροδέκτη 5 (\overline{LDAC}), σε χαμηλό δυναμικό τα δεδομένα του Input Register μεταφέρονται στον DAC Register (βλ. σχήμα 5.11), με αποτέλεσμα μετά από 4.5 μsec (settling time) να έχουμε στον ακροδέκτη 8 την αναλογική τάση. Είναι προφανές πως αν θέλουμε η αναγέννηση της τάσεως εξόδου (Vout) να εξαρτάται μόνο από το \overline{CS} pin, τότε πρέπει να γειώσουμε τον ακροδέκτη 5 (\overline{LDAC}).

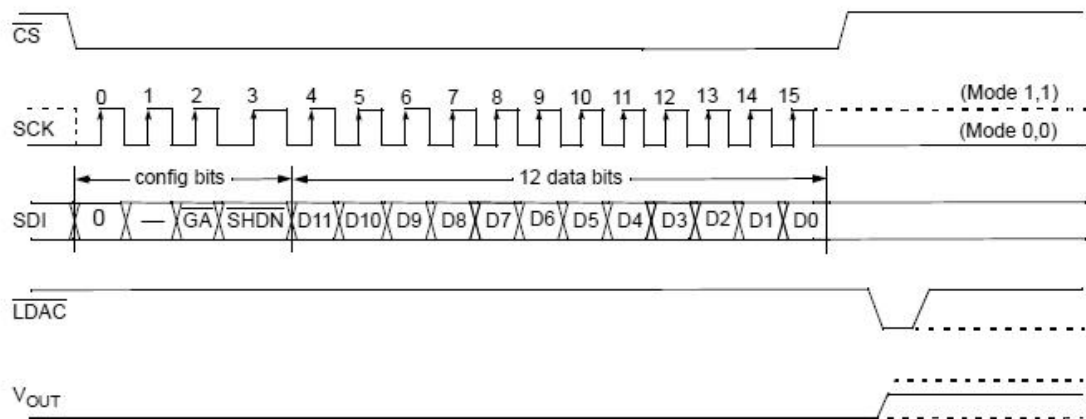
Στο σχήμα 5.13 [7], μπορούμε να δούμε αναλυτικά την δομή των 16bits που πρέπει να στέλνουμε κάθε φορά στο chip μέσω της γραμμής SDI. Το πρώτο bit πρέπει υποχρεωτικά να είναι το 0, διαφορετικά οτιδήποτε στείλουμε μετά θα αγνοηθεί. Το δεύτερο bit μπορεί να είναι οτιδήποτε, ενώ το τρίτο bit καθορίζει το κέρδος του τελεστικού ενισχυτή: Αν είναι 1 τότε το κέρδος είναι 1X, οπότε η μέγιστη τάση εξόδου θα είναι σχεδόν ίση με την τάση αναφοράς, $V_{max} \approx 2.048 \text{ Volts}$, ενώ αν είναι 0 τότε το κέρδος γίνεται 2X με αποτέλεσμα η $V_{max} \approx$

¹ Η διακριτική ικανότητα με σε οποιαδήποτε μηχανική διαστάσεις ορίζεται ως το ελάχιστο σημείο απόδοσης που προκαλεί μεταβολή στο σήμα εξόδου.

² Άξιζε να αναφέρουμε πως ο ελεγκτικός δημοφιλής - αν και σήμερα παρωχημένος - τελεστικός, 741, έχει Skew Rate=0.5 V/μs και έχει χρησιμοποιηθεί ακόμη και σε κονσόλες έχοντας κατά την διακατία του 50



Σχήμα 5.11: Λειτουργικό διάγραμμα του MCP1821



Σχήμα 5.12: Επικοινωνία με το MCP1821

W-x	W-x	W-x	W-0	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
0	—	\overline{GA}	SHDN	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
bit 15								bit 0							

bit 15 ⁽¹⁾ 0 = Write to DAC register
1 = Ignore this command

bit 14 — Don't Care

bit 13 **GA**: Output Gain Selection bit
1 = 1x ($V_{OUT} = V_{REF} * D/4096$)
0 = 2x ($V_{OUT} = 2 * V_{REF} * D/4096$), where internal $V_{REF} = 2.048V$.

bit 12 **SHDN**: Output Shutdown Control bit
1 = Active mode operation. V_{OUT} is available.
0 = Shutdown the device. Analog output is not available. V_{OUT} pin is connected to 500 k Ω (typical).

bit 11-0 **D11:D0**: DAC Input Data bits. Bit x is ignored.

Σχήμα 5.13: Κατεχωρητής εντολής εγγραφής για το MCP4821

Τεχνικά Χαρακτηριστικά του MCP4821 για $V_S=5$ Volts	
Όνομαστική ανάλυση	12 bits
Settling time	4,5 μ sec
Μέγιστη συχνότητα ρολογού	20 MHz
Τάση αναφοράς V_{ref}	2,048 Volts (Εσωτερικά)
Αριθμός καναλιών	1
Πρωτόκολλο Επικοινωνίας	I ² C

Πίνακας 5.2: Μερικά τεχνικά χαρακτηριστικά του MCP4821

4.096 Volts⁴. Με το τέταρτο bit - αν αυτό είναι 0 - μπορούμε να απενεργοποιήσουμε τον μετατροπέα⁴. Τέλος ακολουθούν τα 12 data bits που θέλουμε να μετατραπούν σε αναλογική τάση, από το περισσότερο σημαντικό ψηφίο (bit 11) προς το λιγότερο σημαντικό ψηφίο (bit 0). Στο σημείο αυτό πρέπει να πούμε πως εμείς χρησιμοποιούμε το chip για κέρδος του τελευταίου 1x, με αποτέλεσμα σε κάθε κύκλο να πρώτα 4 bits να είναι η ακολουθία 0011.

Στον πίνακα 5.2 φαίνονται μερικά τεχνικά χαρακτηριστικά του MCP4821. Η τάση εξόδου δίνεται από τον τύπο:

$$V_{out}(Volts) = \frac{2.048 \times D_{12}}{2^{12}} \times G$$

, όπου 2,048V η εσωτερική τάση αναφοράς, $D_{12} \in [0, 4095]$, ο ψηφιακός κωδικός εισόδου με μήκος 12 bits και $G=1$ ή 2, το κέρδος του ενσωματωμένου τελευταίου. Τέλος είναι προφανές πως για $G=1$ η αξία σε τάση εξόδου ενός LSB ψηφίου θα είναι, $V_{ref}/2^{12} = 2.048 \text{ Volts}/4096 = 0.5 \text{ mV}$.

5.3.3 Ο μικροελεγκτής Propeller

Όπως φαίνεται στο σχήμα 5.5, η καρδιά του ψηφιακού υποσυστήματος είναι το Propeller chip. Αυτό είναι υπεύθυνο για την παραγωγή των σημάτων ελέγχου προς τους δύο μετατροπείς, καθώς και για την ψηφιακή επεξεργασία των δειγμάτων, προκειμένου να δημιουργήσουμε το feedback control φίλτρο που απαιτείται. Στην συνέχεια θα αναφερθούμε πολύ συνοπτικά σε κάποια βασικά χαρακτηριστικά του μικροελεγκτή.

³ Για την ακρίβεια θα είναι, $V_{max} = Gain \cdot V_{ref} \times 4095/4096$.

⁴ Το serial interface όμως παραμένει ενεργό με αποτέλεσμα να μπορούμε να επενεργοποιήσουμε ξανά το chip με μια καινούρια εντολή.

Η αρχιτεκτονική του Propeller chip διαφέρει από αυτές άλλων μικροελεγκτών (AVR, PIC). Εδώ δεν έχουμε έναν μικροελεγκτή, αλλά 8 παράλληλους επεξεργαστές (coqs) καθένας από τους οποίους έχει την δική του στατική μνήμη μεγέθους 2 Kb, δύο timers/counters και έναν video generator συνδεδεμένος στον πρώτο timer, όπως βλέπουμε στο σχήμα 5.14 [11]. Κάθε coq μπορεί να εκτελεί 20 MIPS (μέγιστη συχνότητα ρολογιού 80 MHz) και επιπλέον έχει πρόσβαση σε διαμοιραζόμενους πόρους, όπως για παράδειγμα είναι η κοινή για όλους τους επεξεργαστές στατική κύρια μνήμη RAM, μεγέθους 32 Kb (Hub main memory). Επιπλέον υπάρχει και μια μνήμη ROM μεγέθους 32 Kb, στην οποία είναι αποθηκευμένα, το σετ χαρακτήρων, πίνακες για μαθηματικές συναρτήσεις, ο Bootloader και τέλος ο Spin interpreter, ο οποίος είναι ο ενσωματωμένος διερμηνέας για την υψηλού επιπέδου γλώσσα προγραμματισμού Spin.

Ανάλογα με το πότε επιτρέπεται σε κάποιον επεξεργαστή να έχει πρόσβαση σε κάποιον διαμοιραζόμενο πόρο, οι πόροι αυτοί χωρίζονται σε δύο κατηγορίες:

- Στους κοινούς πόρους όπως είναι για παράδειγμα τα I/O pins και ο System counter. Τους πόρους αυτούς μπορεί να τους χρησιμοποιήσει οποιοδήποτε coq, σε οποιαδήποτε χρονική στιγμή.
- Στους αρμοίβια αποκλειόμενους πόρους, όπως είναι η κύρια μνήμη RAM των 32 Kb (hub main mem) και οι configuration registers, οι οποίοι ελέγχονται από το Hub και τον δίαυλο του. Πρόσβαση σε τέτοιους πόρους έχουν όλα τα coqs αλλά σε καθορισμένα time slots, ακολουθώντας μια round-robin αρχιτεκτονική. Επειδή έχουμε 8 coqs θα έπρεπε κανονικά να επιτρέπεται η πρόσβαση σε έναν τέτοιου πόρο για κάποιο coq, 1 φορά για κάθε 8 κύκλους ρολογιού. Επειδή όμως η συχνότητα που λειτουργεί ο δίαυλος αυτός είναι η μισή του System clock, αυτό συνεπάγεται πρόσβαση μια φορά για κάθε 16 κύκλους ρολογιού, για κάποιο coq.

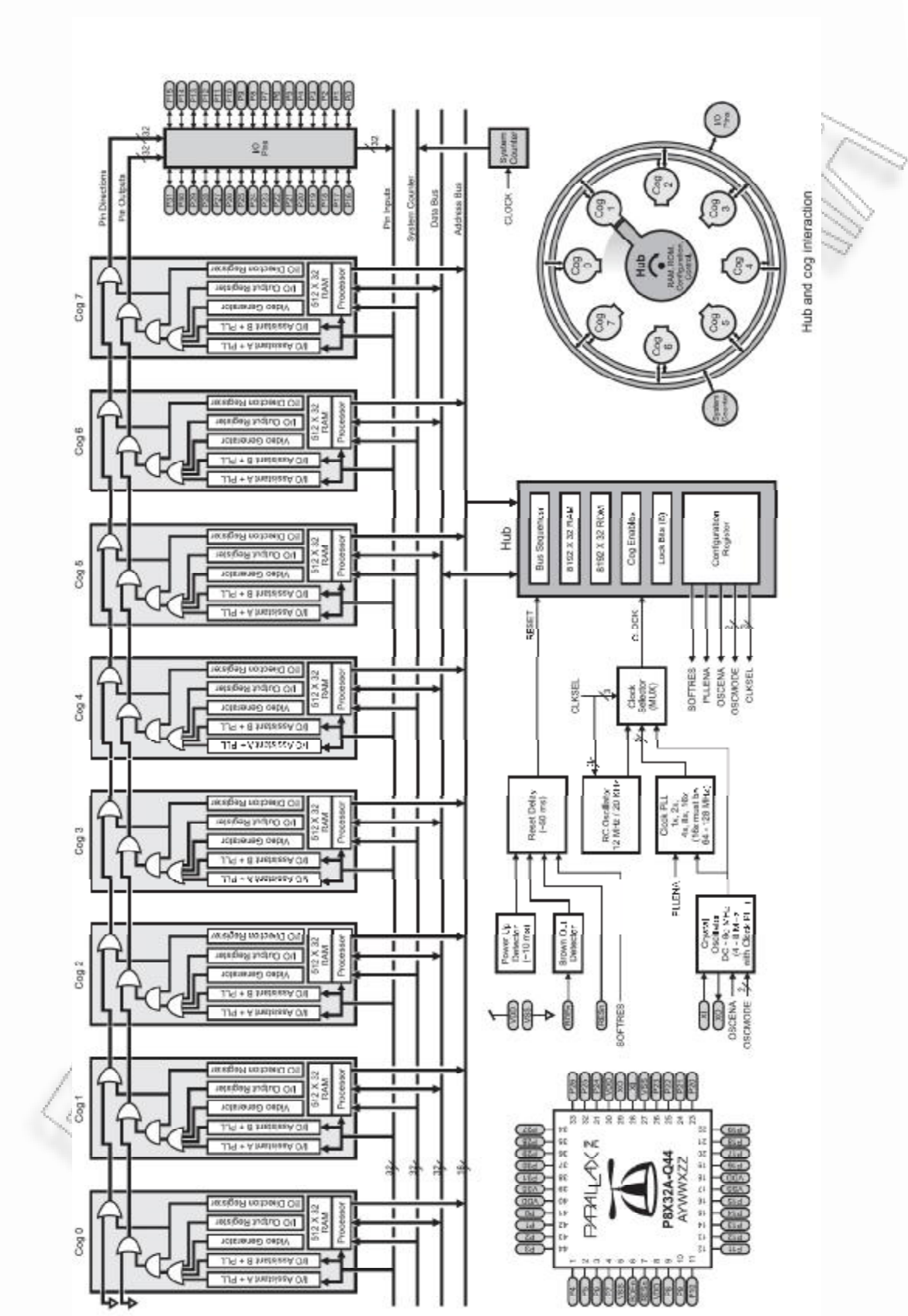
Κάθε επεξεργαστής εκτελεί εντολές πάνω σε δεδομένα από την εσωτερική γρήγορη μνήμη του, των 2 Kb, η οποία είναι οργανωμένη σε κελιά των 32 bits. Εδώ λοιπόν δεν έχουμε μια αφιερωμένη ειδική μνήμη που αποθηκεύονται οι εντολές και μια ξεχωριστή μνήμη για τα δεδομένα, αλλά μια κοινή μνήμη και για τα δύο, μεγέθους 2 Kb (αρχιτεκτονική "von - Neumann" ³). Από τα παραπάνω είναι προφανές πως κάθε εντολή έχει μέγεθος 32 bits (όπως είναι και το μήκος για κάθε δεδομένο), γεγονός που την κάνει πολύ περισσότερο ισχυρή από τις κοινές 8 bit εντολές που έχουν άλλοι ανταγωνιστικοί μικροελεγκτές. Την εσωτερική αυτή μνήμη των 2 Kb μπορούμε λοιπόν να την δούμε σαν 512 το πλήθος καταχωρητές μήκους 32 bits ο καθένας, όπου εκεί αποθηκεύονται οι εντολές και τα δεδομένα του προγράμματός μας. Από αυτούς τους 512 καταχωρητές διαθέσιμοι στον χρήστη είναι οι 496 (0 - 495), αφού οι τελευταίοι 16 (496 - 511) χρησιμοποιούνται για την πρόσβαση, κάθε επεξεργαστή στα I/O pins, στους δύο τοπικούς timers και στον κεντρικό system counter, όπως βλέπουμε στο σχήμα 5.15 [11].

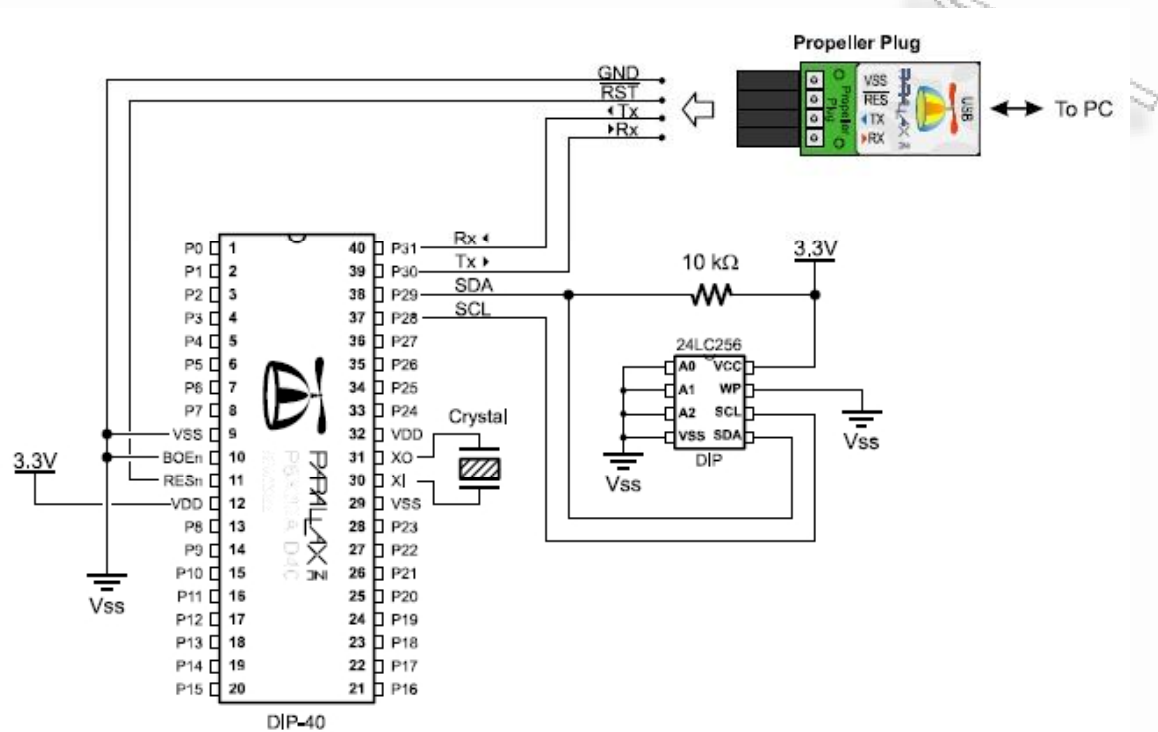
Όπως αναφέραμε, μια εντολή σε γλώσσα Propeller Assembly αποτελείται από 32bits καταναμημένα ως εξής:

- Instruction: Κωδικός εντολής (bits 31:26)
- Effects: Ενεργοποίηση flags και αποτελέσματος, μετά την εκτέλεση (Z, C, R) (bits: 25:23)
- Immediate addressing: Άμεση διευθυνσιοδότηση (bit: 22)
- Condition: Υπό συνθήκη εκτέλεση (bits: 21:18)
- Destination: Καταχωρητής προορισμού (bits 17:5)
- Source: Καταχωρητής πηγή ή άμεση τιμή μεγέθους 9bits (bits 8:0)

Παρατηρώντας την παραπάνω δομή μιας τυπικής assembly εντολής, μπορούμε να καταλάβουμε πόσο πιο ισχυρή είναι μια τέτοια εντολή από κάποια που θα διαθέτα για παράδειγμα μόνον 8 bits, όπως συμβαίνει σε άλλους μικροελεγκτές. Βλέπουμε για παράδειγμα πως υπάρχει η δυνατότητα να εκτελεστεί μια εντολή ανάλογα με το αν αληθεύει ή όχι κάποια συνθήκη (conditional execution) χωρίς επιβάρυνση στον χρόνο εκτέλεσης! Η συνθήκη αυτή αφορά το περιεχόμενο των (δύο ακριβώς) flags που διαθέτει ο μικροελεγκτής: Zero flag και Carry flag. Έτσι ανάλογα με το περιεχόμενο των 2 αυτών flags μια εντολή μπορεί να εκτελεστεί ή όχι. Επίσης έχουμε την δυνατότητα μια εντολή αφού εκτελεστεί να επηρεάσει ή όχι τα δύο αυτά flags (WC: Write Carry, WZ: Write Zero), καθώς και να γράφει ή όχι το αποτέλεσμα στον καταχωρητή προορισμού (WR: Write Result, NR: No Result), παρακάμπτοντας την εξόρισμό συμπεριφορά της. Είναι φανερό λοιπόν πως οι δύο αυτές δυνατότητες, καθιστούν την γλώσσα assembly πανίσχυρη. Συνοψίζοντας, μια εντολή assembly έχει την ακόλουθη σύνταξη: <Label> <Condition> Instruction Operands <Effects>

³Σε αντίθεση με το Propeller οι μικροελεγκτές PIC και AVR έχουν ξεχωριστή μνήμη για τις εντολές, ακολουθώντας την αρχιτεκτονική τύπου "Harvard".

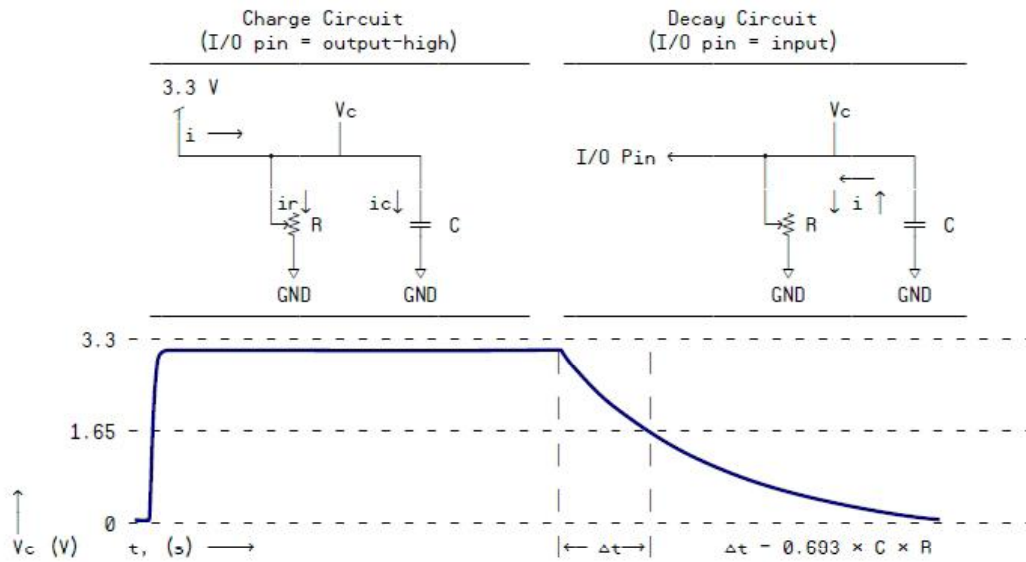




Σχήμα 5.16: Παράδειγμα συνδεσμολογίας του Propeller chip με εξωτερική μνήμη EEPROM και ένα PC

Τεχνικά Χαρακτηριστικά του Propeller chip	
Μοντέλο	P8X32A
Τάση λειτουργίας	3.3 Volts
Μεγιστή συχνότητα ρολογιού	80 MHz
Αριθμός παράλληλων επεξεργαστών (Cogs)	8
Υπολογιστική ισχύς	20 MIPS/Cog, για συχνότητα λειτουργίας 80 MHz
Κύρια μνήμη RAM/ROM	64 Kb (32 Kb RAM + 32 Kb ROM)
Cog RAM	512 × 32 bits για κάθε Cog
I/O pins	32 CMOS level pins, με input threshold=1.65 Volts
Κατανάλωση	500 μΑ/MIPS

Πίνακας 5.3: Μερικά τεχνικά χαρακτηριστικά του μικροελεγκτή Propeller



Σχήμα 5.17: Μέτρηση του χρόνου εκφόρτισης πυκνωτού μέσω μεταβλητής αντιστάσεως

εος μνήμη, όπου έχουμε πρόσβαση μόνο σε επίπεδο long (32 bits). Επιπλέον τα δεδομένα στην κύρια μνήμη αποθηκεύονται με τρόπο να LSB και τελευταίο bit MSB, δηλαδή το MSB αποθηκεύεται πιο χαμηλά στην μνήμη, αφού ο μικροελεγκτής είναι αρχιτεκτονικής Little Endian. Τέλος όσον αφορά την κωδικοποίηση των αρνητικών ακεραίων, χρησιμοποιείται η παράσταση συμπληρώματος ως προς 2 (2's complement). Στον πίνακα 5.3, συνοψίζουμε ορισμένα τεχνικά χαρακτηριστικά του Propeller chip.

5.3.4 Υποσύστημα ελέγχου των παραμέτρων Feedback gain, Delay time.

Προκειμένου ο χρήστης να μπορεί να θέτει τις δύο παραμέτρους ελέγχου του συστήματος, χρησιμοποιήσαμε δύο περιστροφικές μεταβλητές αντιστάσεις (ποτενσιόμετρα). Έτσι ανάλογα με την γωνιακή μετατόπιση κάθε ποτενσιόμετρου μεταβάλλεται η αντίσταση μεταξύ δύο ακροδεκτών του, από μια ελάχιστη τιμή 0 ohms μέχρι $R1$ ohms που είναι και η ονομαστική αντίστασή του. Κατά μια έννοια, η μεταβλητή αντίσταση είναι ένας ωμικός μεταλλάκτης μετατόπισης, αφού η θέση που βρίσκεται κάθε φορά μετατρέπεται σε κάποια τιμή ωμικής αντίστασης.

Προκειμένου να μετρήσουμε αυτήν την αντίσταση ψηφιακά, έχουμε δύο δυνατότητες. Η μια είναι να συνδέσουμε στα δύο άκρα του ποτενσιόμετρου μια συνεχή τάση έτσι ώστε ανάλογα με την μετατόπιση να παίρνουμε μεταβλητή τάση αντί για μεταβλητή αντίσταση (διαρέτης τάσης), και στην συνέχεια να συνδέσουμε τον τρίτο ακροδέκτη του στην είσοδο ενός A-D μετατροπέα. Ανάλογα με την θέση του ποτενσιόμετρου μεταβάλλεται τώρα η τάση μεταξύ του τρίτου ακροδέκτη και της γης, με αποτέλεσμα να παίρνουμε μέσω του A-D μετατροπέα μια ψηφιακή τιμή.

Ο δεύτερος τρόπος είναι τελείως ανέξοδος, μιας και τα μόνο που χρειάζεται είναι ένας πυκνωτής κατάλληλης τιμής. Η αρχή λειτουργίας του βασίζεται στην μέτρηση του χρόνου εκφόρτισης του πυκνωτού C, μέσω της μεταβλητής αντιστάσεως R, όπως φαίνεται στο σχήμα 5.17 [5].

Βλέπουμε πως η μία άκρη του πυκνωτού συνδέεται απευθείας σε κάποιο ακροδέκτη του μικροελεγκτή. Έτσι θέτοντας τον ακροδέκτη αυτόν σε υψηλό δυναμικό, ο πυκνωτής αρχίζει να φορτίζεται (σχεδόν ακαριαία) αποκτώντας τάση 3.3 Volts στα άκρα του. Στην συνέχεια μετατρέπουμε τον ακροδέκτη αυτόν σε input με αποτέλεσμα ο πυκνωτής να αρχίσει να εκφορτίζεται μέσω της μεταβλητής αντιστάσεως R. Αποδεικνύεται πως ο χρόνος που απαιτείται για να γίνει η τάση στα άκρα του πυκνωτού η μισή (1.65 V), δίνεται από την σχέση [5]:

$$\Delta t = 0.693 \times C \times R$$

Συνεπώς ο χρόνος Δt είναι ανάλογος της μεταβλητής αντιστάσεως R. Το μόνο που μένει είναι να βρούμε έναν τρόπο να μετρήσουμε τον χρόνο αυτόν μέσω του μικροελεγκτή. Στο σημείο αυτό μπορεί να μας βοηθήσει ο ένας από τους δύο ανεξάρτητους μετρητές που έχει ενσωματωμένους κάθε κομ, όπως αναφέραμε στην ενότητα 5.3.3.

Ένας τέτοιος μετρητής, για παράδειγμα ο A, μπορεί να τεθεί σε λειτουργία "Positive Detector" που σημαίνει

πως σε κάθε κύκλο ρολογιού θα προστίθεται το περιεχόμενο του καταχωρητή FRQA στον καταχωρητή PIISA, για όσο χρόνο κάποιος ακροδέκτης³ βρίσκεται σε υψηλό δυναμικό. Στο σημείο αυτό πρέπει να υπενθυμίσουμε πως όταν ένα pin έχει τάση κάτω από $V_{DD}/2=1.65\text{ V}$, τότε θεωρείται ότι βρίσκεται σε χαμηλό δυναμικό (βλ. πίνακα 5.3). Έτσι αρχικά εμείς θέτουμε FRQA=1 και μηδενίζουμε τον καταχωρητή PIISA (PIISA=0), αμέσως μετά αφού έχουμε μετατρέψει τον ακροδέκτη του Propeller που συνδέεται με τον πυκνωτή, από output σε input. Προφανώς η τιμή που θα έχει αποθηκευτεί στον καταχωρητή PIISA όταν ολοκληρωθεί η εκφόρτιση στα 1.65 V, θα είναι ο χρόνος Δt σε κύκλους ρολογιού.

Ένα σοβαρό πρόβλημα που προκύπτει από την χρήση του κυκλώματος του σχήματος 5.17, είναι το πολύ μεγάλο ρεύμα που αναπτύσσεται κατά την φόρτιση του πυκνωτού⁴. Ο χρόνος που συμβαίνει αυτό είναι πολύ λίγος, αλλά επειδή αυτό επαναλαμβάνεται σε κάθε IOP, υπάρχει ενδεχόμενο να καταστραφεί κάποια στιγμή ο ακροδέκτης του μικροελεγκτή που συνδέεται με τον πυκνωτή⁵. Επιπλέον αυτές οι απότομες εξάρσεις του ρεύματος μπορούν να προκαλέσουν τάρξια στο ακουστικό σήμα, αλλάζοντας τον ήχο. Για να αποφύγουμε λοιπόν τα φαινόμενα αυτά μπορούμε να παρεμβάλουμε μεταξύ του ακροδέκτη του μικροελεγκτή και του πυκνωτή, μια αντίσταση R1 που έχει ως στόχο την μείωση του ρεύματος φόρτισης, όπως φαίνεται στο σχήμα 5.18.

Η προσθήκη της αντιστάσεως R1 καταργεί την γραμμικότητα μεταξύ του χρόνου εκφόρτισης Δt και της τιμής του ποτενσιόμετρου R_x , αν η τιμή της δεν εκλεγεί κατάλληλα. Πράγματι αν αναλύσουμε το βελτιωμένο κύκλωμα του σχήματος 5.18, θα δούμε πως η τάση που θα αναπτυχθεί στο τέλος της φόρτισης του πυκνωτού δεν θα είναι σταθερή όπως πριν (3.3 V), αλλά θα εξαρτάται από την μεταβλητή αντίσταση του ποτενσιόμετρου R_x , σύμφωνα με την ακόλουθη σχέση:

$$V_f = \frac{3.3R_x}{R_x + R_1} \quad (5.1)$$

Συνέπεια αυτού είναι ο χρόνος εκφόρτισης στα 1.65 V να δίνεται προς:

$$\Delta t = CR_x \left(0.693 + \ln \frac{R_x}{R_x + R_1} \right)$$

Σύμφωνα με τις δύο τελευταίες σχέσεις, για να διατηρηθεί η γραμμικότητα μεταξύ του χρόνου Δt και της ένδειξης του ποτενσιόμετρου R_x , θα πρέπει να εκλέξουμε μια τιμή για την R1 κατά πολύ μικρότερη από την ονομαστική αντίσταση του ποτενσιόμετρου. Μια παρήχηρηση ο χειρική με αυτό είναι ότι μια επιπόλαια εκλογή της τιμής R1, μπορεί να οδηγήσει στην μείωση του μήκους της ωφέλιμης διαδρομής του ποτενσιόμετρου, αφού για τιμές της $R_x \leq R_1$, δεν υφίσταται χρόνος εκφόρτισης Δt στα 1.65 V, από την στιγμή που η τάση φόρτισης στην περίπτωση αυτή θα είναι μικρότερη ή ίση των 1.65 V (εξίσωση 5.1).

Όπως βλέπουμε στην συνδεσμολογία του ψηφιακού υποσυστήματος στο σχήμα 5.19, προκειμένου να ελέγχει ο χρήστης τις παραμέτρους Feedback gain και Delay time, χρησιμοποιήσαμε δύο όμοια περυστροφικά γραμμικά ποτενσιόμετρα ονομαστικής αντιστάσεως 10 K Ω . Ο πυκνωτής που συνδέεται στο ποτενσιόμετρο του Feedback gain έχει χωρητικότητα 267 nF⁶, ενώ αυτός του Delay time ποτενσιόμετρου, 330 nF. Επίσης συνδέσαμε το κάθε ένα από τα δύο αυτά κυκλώματα με τους ακροδέκτες P11 και P15 του Propeller chip αντίστοιχως, μέσω αντιστάσεως R1=1K για τους λόγους που αναφέρθηκαν προηγουμένως.

Σε προηγούμενη παράγραφο αναλύσαμε τον τρόπο χρήσης ενός μετρητή για κάποιο cos, προκειμένου να μετρήσουμε να μετρήσουμε τον χρόνο εκφόρτισης ενός πυκνωτή μέσω αντιστάσεως. Είδαμε πως στον καταχωρητή PIISA (αν πρόκειται για τον πρώτο counter A) αποθηκεύεται ο χρόνος αυτός σε clock ticks, δηλαδή κάθε μια μονάδα θα αντιστοιχεί σε χρόνο 12.5 ns ($f_{clk} = 80\text{ MHz}$). Αν τώρα διαφέρουμε το περιεχόμενο του καταχωρητή PIISA με το 512, τότε ο χρόνος που θα αντιστοιχεί σε κάθε καινούργια μονάδα θα είναι $12.5\text{ ns} \times 512 = 6.4\text{ }\mu\text{sec}$. Αυτό ακριβώς εφαρμόσαμε στον κώδικα που αναπτύξαμε σχετικά με το διάβασμα των δύο ποτενσιόμετρων μέσω των counters του Propeller chip (βλ. Ψευδοκώδικα 1). Έτσι οι τιμές που προκύπτουν από την περιγραφή των ποτενσιόμετρων είναι για το δεν Delay pot από 16 - 385, ενώ

³ Για το παρόν άρθρο μας εννοούμε τον ακροδέκτη που συνδέεται με τον πυκνωτή.

⁴ Μέχρι ο πυκνωτής να φορτιστεί, συμπεριφομοποιείσιν βραχυκύκλωμα στα πάντες πολύ μεγάλο θερμοκρασιακό άραγμα από τον ακροδέκτη του μικροελεγκτή.

⁵ Άλλοτεσι οδικα για το Propeller chip, υπάρχει προστασιατικη διαδος σε κάθε ακροδέκτη και η τάση μείναι το ρεύμα φόρτισης, αποπέταται την καταστροφή του.

⁶ Η τιμή αυτή προκύπτει από την εν παραλλήλως σύνδεση δύο πυκνωτών με χωρητικότητας 220 nF και 47 nF αντίστοιχα.

για το Feedback pot από 16 – 293, όπου η κάθε μονάδα αντιστοιχεί σε χρόνο ίσο προς 6.4 μsec¹². Τέλος να σημειώσουμε πως ο λόγος που διατρέσαμε τις μετρούμενες τιμές με αποτέλεσμα την μείωση της ανάλυσης, έχει να κάνει με την λογική ισοπέδηση μας από το μερρητικό αυτό σύστημα, να παράγει την ίδια τιμή κάθε φορά που εκτελείται το ίδιο "πείραμα". Έτσι μετά από δικούς μας πειραματισμούς, καταλήξαμε στην τελική ανάλυση των 9 bits σχετικά με τους χρόνους εκφόρτισης, για τις δεδομένες χωρητικότητες και αντιστάσεις.¹³

5.3.5 Λογισμικό ανάπτυξης

Οι γλώσσες προγραμματισμού που χρησιμοποιήσαμε για τον προγραμματισμό του Propeller chip, είναι η PropBASIC (Version 1.03) και η Assembly. Η γλώσσα PropBASIC είναι ένας μεταγλωττιστής της γλώσσας BASIC για τον μικροελεγκτή Propeller, ο οποίος συνεργάζεται με το περιβάλλον ανάπτυξης λογισμικού Bst Tool IDE. Και τα δύο αυτά προγραμματιστικά εργαλεία διατίθενται δωρεάν από τις διευθύνσεις, <http://www.fnarfbargle.com/PropBasic/> και <http://www.fnarfbargle.com/bst.html> αντίστοιχα.

Η PropBASIC είναι ένας inline μεταγλωττιστής, ο οποίος μεταφράζει κάθε δήλωση της γλώσσας BASIC σε ένα block από εντολές Assembly, χωρίς να κάνει κάποια βελτιστοποίηση του κώδικα. Ο χρήστης μπορεί να δει με έναν έξοπνο τρόπο μέσα από το περιβάλλον Bst Tool IDE, τον παραγόμενο Assembly κώδικα δίπλα από κάθε γραμμή μας δήλωσης σε γλώσσα PropBASIC. Επίσης η γλώσσα μας δίνει, την δυνατότητα να παρεμβάλλουμε στον BASIC κώδικά μας και αυτούσιες εντολές Assembly. Με τον τρόπο αυτό έχουμε έναν τρόπο να παράγουμε εύκολα, γρήγορα και κατανοητό Assembly κώδικα, η χρήση του οποίου είναι απαραίτητη για κρίσιμες από πλευράς χρόνου εφαρμογές, όπως είναι το σύστημα που αναπτύξαμε.

Η δομή του προγράμματος που αναπτύξαμε σε σχέση με την αρχιτεκτονική του Propeller φαίνεται στο σχήμα 5.20. Το πρόγραμμά μας χρησιμοποιεί μόνο δύο (Cog0, Cog1) από τους οκτώ διαθέσιμους επεξεργαστές του Propeller. Αρχικά ο πρώτος επεξεργαστής, Cog0, αναθέτει στον δεύτερο επεξεργαστή, Cog1, το διάγραμμα των παραμέτρων ελέγχου, Delay time και Feedback gain μέσω των αντίστοιχων ποτενσιομέτρων. Η διαδικασία αυτή πραγματοποιείται από τους δύο ενσωματωμένους μετρητές, COUNTERA και COUNTERB του Cog1, όπως αναλύσαμε στην υποενότητα 5.3.4. Εδώ πρέπει να υπενθυμίσουμε πως ο επεξεργαστής Cog1 από την στιγμή που ενεργοποιήθηκε αρχίζει να λειτουργεί τελείως ανεξάρτητα και παράλληλα με το Cog0.¹⁴

Οι δύο ανεξαρτητές επικοινωνούν μεταξύ τους μέσω των δύο μεταβλητών ParDly και ParFb της κύριας μνήμης RAM, όπως φαίνεται χαρακτηριστικά στο σχήμα 5.20. Έτσι ο επεξεργαστής Cog1 ανανεώνει το περιεχόμενο των μεταβλητών αυτών όταν υπάρξει κάποια αλλαγή στα ποτενσιομέτρα ελέγχου από τον χρήστη και ο επεξεργαστής Cog0, τις διαβάσει σε κάθε loop. Επειδή τώρα υπάρχει το ενδεχόμενο, παρόλο που ένα ποτενσιομέτρο να είναι ακίνητο η τιμή που διαβάσει το Cog1 από αυτό να αμφιταλαντεύεται μεταξύ δύο γειτονικών τιμών, έχουμε προβλέψει στον κώδικά μας (ψευδοκώδικας 1), το Cog1 να ανανεώνει τις μεταβλητές ParDly και ParFb μόνον αν έχουμε αλλαγή κατά απόλυτο τιμή μεγαλύτερη αυτές ανάλογα με την θέση των ποτενσιομέτρων είναι, $ParDly = 512, 544, 576, \dots, 12320$ και $ParFb = 16, 17, 18, \dots, 293$.

Συνεχίζοντας, μετά από την ενεργοποίηση του Cog1, το Cog0 μπαίνει σε έναν ατέρμονα βρόχο, όπου εκτελούνται διαδοχικά οι ακόλουθες εργασίες:

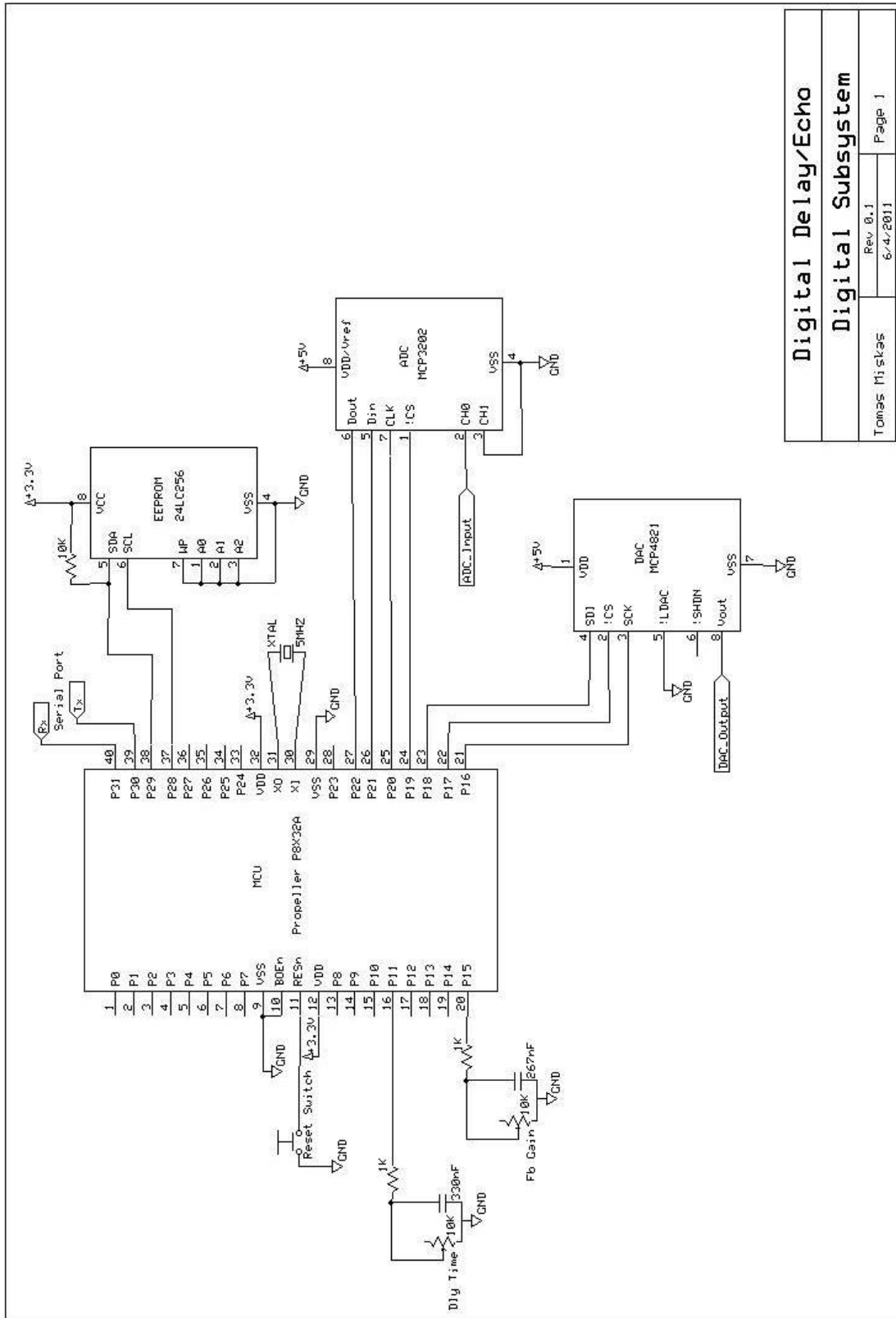
- Επικοινωνία με τον A-D μετατροπέα, MCP9202 (ADC Section).
- Αλγόριθμος Echo (Echo Section).
- Επικοινωνία με τον D-A μετατροπέα, MCP1821 (DAC Section).

Προκειμένου να υλοποιήσουμε ένα feedback comb φίλτρο, χρειαζόμαστε την δυνατότητα να μπορούμε να αποθηκεύουμε παρμένα δείγματα του σήματος εξόδου. Για τον λόγο αυτό δεσμεύουμε ένα μεγάλο ποσό της κύριας μνήμης Ram - περίπου 26 Kb - υπό μορφή μνήμης με όνομα Buf, όπου μπορούμε να αποθηκεύσουμε 13000 δείγματα μήκους 16 bits. Το ποσό της μνήμης που θα χρησιμοποιούμε σε κάθε κύκλο εξαρτάται από την μεταβλητή ParDly που συνδέεται με το ποτενσιομέτρο Delay time. Η μνήμη αυτή πρέπει να έχει κυκλική δομή (circular buffer), έτσι ώστε όταν φτάνουμε στο τέλος της, αυτή να μην εξαντλείται, αλλά να υπάρχει πάλι πρόσβαση για ανάγνωση-εγγραφή στην αρχή της. Για να μπορούμε να φτιάξουμε μια τέτοια δομή χρειαζόμαστε έναν αυξανόμενο δείκτη, pointer, ο οποίος όταν θα φτάνει την τιμή buflen (buffer length) που ισούται με την παράμετρο ParDly, θα μηδενίζεται. Ο ψευδοκώδικας 2 περιγράφει, αναλυτικά τον αλγόριθμο Echo που εκτελεί ο πρώτος επεξεργαστής, Cog0.

¹² Σε κάθε τιμή υπάρχει offset=16, προκειμένου ακόμη και αν το ποτενσιομέτρο βρίσκεται στην χαμηλότερη του θέση να παράγει και κάποιο παίξιμο (αξιοσημείωτη τιμή).

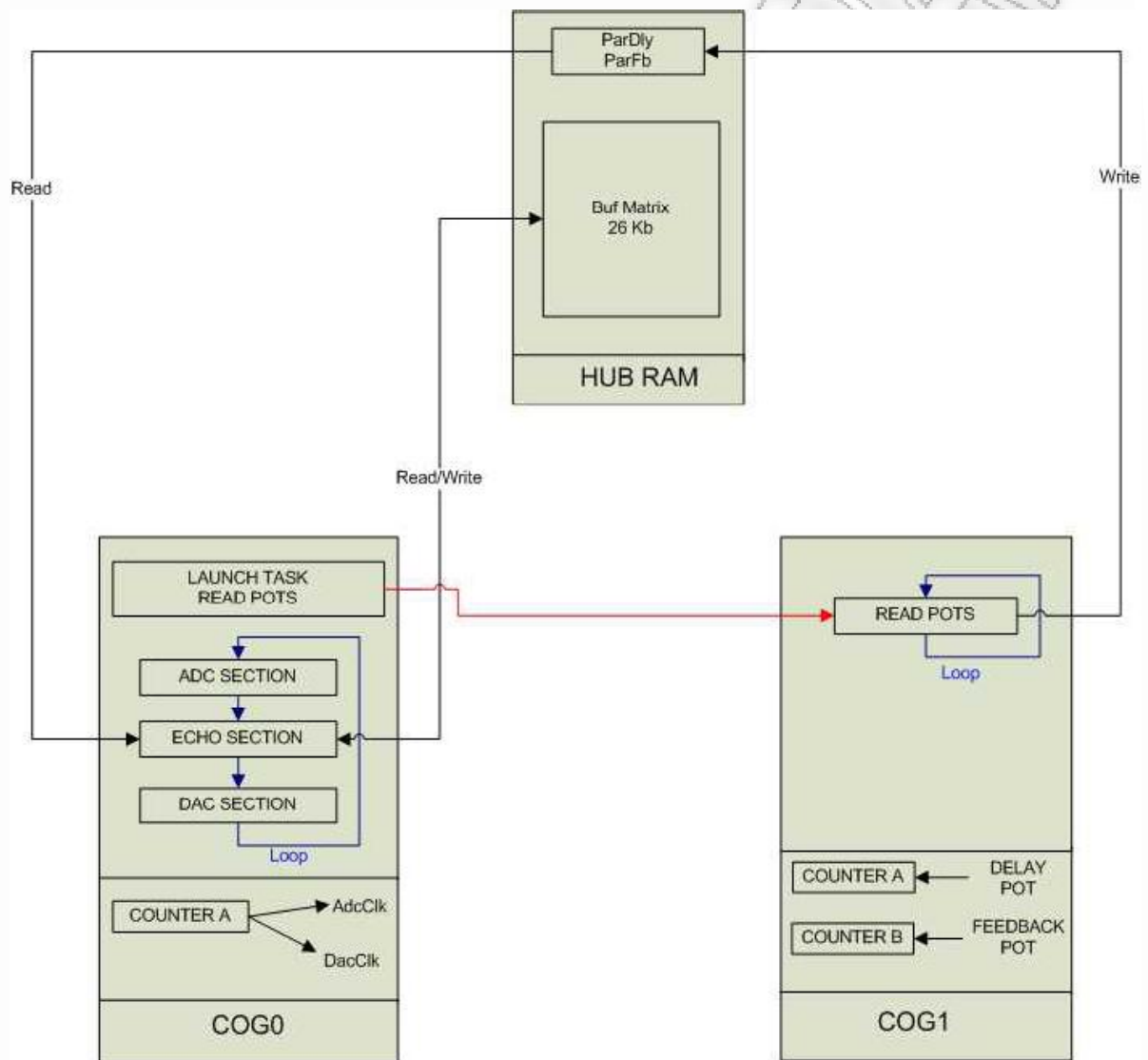
¹³ Στο σημείο αυτό πρέπει να υπενθυμίσουμε πως ο τελικός στόχος μας δεν ήταν η ακριβής μέτρηση του χρόνου εκφόρτισης, αλλά η επίδραση ενός αποθηκευμένου ημι-απείρου διαγρήματος των δύο περιερίτρων ελέγχου του συστήματος.

¹⁴ Αν είχαμε έναν ανεξαρτητικό πλέον συμβολικό μικροελεγκτή θα κατανοούσαμε αναλογιστικά στην χρήση interrupts προκειμένου να διαβάζαμε τα ποτενσιομέτρα με αποτέλεσμα την αύξηση της πολυπλοκότητας του προγράμματος μας.



Digital Delay/Echo	
Digital Subsystem	
Τόμας Πιτσκάς	Rev. 0.1 6/4/2011
	Page 1

Σχήμα 5.19: Ηλεκτρονικό κύκλωμα του ψηφιακού ηχοεπιπέδου



Σχήμα 5.20: Δομή του προγράμματος σε σχέση με το χρησιμοποιούμενο hardware

Ψευδοκώδικας 1 Διάβασμα των ποτενσιομέτρων από το Cog1 (Task: ReadPots)

```

COUNTERA(PosDetector, DlyPin), COUNTERB(PosDetector, FbPin) {Θέσε τους μετρητές CO-
UNTERA και COUNTERB σε λειτουργία "Positive Detector"}
prevDlyTicks = 0, prevFbTicks = 0 {Στις δύο αυτές μεταβλητές αποθηκεύουμε τις μετρήσεις του αμέ-
σως προηγούμενου κύκλου}
loop
  PinDlyPot ← HIGH, PinFbPot ← HIGH
  WAIT(2ms) {Φορτίζουμε τους δύο πυκνωτές για 2 ms}
  PinDlyPot ← INPUTSTATE, PinFbPot ← INPUTSTATE {Μετάτροπή των δύο pins του
Propeller που συνδέονται με τα ποτενσιόμετρα σε κατάσταση INPUT}
  phsa = 0, phsb = 0 {Αρχικοποίηση των δύο μετρητών. Ο phsa αφορά το Delay Pot, ενώ ο phsb το
Feedback pot}
  WAIT(4ms) {Αναμονή για 4 ms προκειμένου η τάση στους 2 πυκνωτές να είναι σίγουρα κάτω από 1.65
V}
  ticks ← phsa/512 {Ο phsa αφορά το ποτενσιόμετρο Delay time}
  ticks ← ticks + 16 {Το εύρος των μετρηθέντων τιμών για τα δεδομένα C,R του συστήματός μας θα είναι:
16 ≤ ticks ≤ 385}
  Delay ← ticks × 32 {Delay = 512, 544, 576, ..., 12320}
  if |ticks - prevDlyTicks| > 1 then {Ανανέωσε την παράμετρο ParDly στην κύρια μνήμη μόνο αν έχουμε
κίνηση του ποτενσιόμετρου}
    ParDly ← Delay
  end if
  prevDlyTicks ← ticks
  ticks ← phsb/512 {Ο phsb αφορά το ποτενσιόμετρο Feedback gain}
  Feedback ← ticks + 16 {Το εύρος των μετρηθέντων τιμών για τα δεδομένα C,R του συστήματός μας θα
είναι: 16 ≤ Feedback ≤ 293}
  if |ticks - prevFbTicks| > 1 then {Ανανέωσε την παράμετρο ParFb στην κύρια μνήμη μόνο αν έχουμε
κίνηση του ποτενσιόμετρου}
    ParFb ← Feedback
  end if
  prevFbTicks ← ticks
end loop

```

Ψευδοκώδικας 2 Ο εκτελούμενος αλγόριθμος Echo από το Cog0

```

COGSTART(ReadPots) {Ανάθεση στο Cog1 το διάβασμα των ποτενσιομέτρων ελέγχου Delay time και
Feedback gain}
pointer = 0, Buf(k) = 0  $\forall k \in [0, 12999]$  {Αρχικοποίηση μεταβλητών και μητρών}
loop
  ADC(Sample) {Πάρε ένα νέο μη προσημασμένο 12 bit δείγμα, sample, από το ADC}
  sample ← sample - 2048 {Μετατροπή του δείγματος sample σε προσημασμένο ακέραιο 12 bit}
  buflen ← ParDly, feedback ← ParFb {Διάβασε από την κύρια μνήμη τις παραμέτρους ParDly, ParFb}

  old ← Buf(pointer) {Διάβασε το (παλιό) δείγμα που βρίσκεται στην θέση pointer της μνήμης και απο-
θήκευσε το στην μεταβλητή old}
  old ← old × feedback/512 {Υπολόγισε ένα κλάσμα του δείγματος old}
  mix ← sample + old {Μίξε το τρέχον δείγμα sample με το ποσοστό του παλιού δείγματος που
υπολόγισες πριν}
  Buf(pointer) ← mix {Γράψε το μεξारισμένο δείγμα mix στην θέση pointer της μνήμης}
  pointer ← pointer + 1
  pointer ← pointer MODULO buflen {Η δομή της μνήμης πρέπει να είναι κυκλική}
  if mix > 2047 then
    mix = 2047
  end if
  if mix < -2048 then
    mix = -2048
  end if
  mix ← mix + 2048 {Μετατροπή της μεταβλητής mix, σε μη προσημασμένο ακέραιο 12bit}
  DAC(mix) {Προώθησε το επεξεργασμένο 12 bit δείγμα mix στο DAC}
end loop

```

Αν παρατηρήσουμε τον ψευδοκώδικα 2 θα δούμε πως σε κάποιο βήμα του αλγορίθμου υπολογίζεται ένα κλάσμα του αποθηκευμένου δείγματος old, ανάλογο με την μεταβλητή feedback που διαβάσαμε από το ποτενσιομέτρο feedback gain. Οι πράξεις που πρέπει να εκτελεστούν είναι ένας πολλαπλασιασμός και μια διαίρεση: $old \times feedback/512$. Ωστόσο στις πράξεις αυτές υπάρχει προτεραιότητα: Πρώτα πρέπει να εκτελεστεί ο πολλαπλασιασμός της (προσημασμένης) μεταβλητής old, μήκους 16 bits, με την (μη προσημασμένη) μεταβλητή feedback, μήκους 9 bit και στην συνέχεια το αποτέλεσμα - που θα έχει μέγιστο μήκος 25 bits - θα διαφεθεί με το $512 = 2^9$. Προφανώς το τελικό αποτέλεσμα θα έχει μήκος 16 bits. Έτσι προκειμένου να πετύχουμε τον πολλαπλασιασμό μιας ακέραιας τιμής - που παριστάνει ένα αποθηκευμένο δείγμα - με μια τιμή μικρότερη της μονάδας - η οποία παριστάνει το κέρδος της ανόδρασης - θα πρέπει να καταφύγουμε σε αυτό το τέχνησμα. Επειδή η μεταβλητή feedback παίρνει τιμές από 16 έως 293, το ισοδύναμο εύρος του κέρδους της ανόδρασης (feedback gain) θα είναι από $16/512 \approx 0.03$ έως $293/512 \approx 0.57$.

Ένα ζήτημα που προέκυψε σχετικά με την πραγματοποίηση των παραπάνω αριθμητικών πράξεων, αφορούσε την απουσία hardware multiplier από το Propeller chip. Για τον λόγο αυτό τροποποιήσαμε τον κώδικα πολλαπλασιασμού σε assembly που προτείνεται στην σελίδα 380 του Propeller Manual V1.1 (βλέπε αναφορά [1]), προκειμένου να εκτελεστεί ο απαιτούμενος πολλαπλασιασμός μεταξύ μιας προσημασμένης ακέραιας μεταβλητής μήκους 16bits με μια μη προσημασμένη ακέραια μεταβλητή μήκους 9 bits⁵. Ο ψευδοκώδικας 3 περιγράφει τον αλγόριθμο που εφαρμόσαμε. Να σημειώσουμε πως για την υλοποίηση σε κώδικα της επαναληπτικής διαδικασίας του πολλαπλασιασμού, δεν χρειάζεται η αποθήκευση των ενδιάμεσων αποτελεσμάτων, $Y^{[k]}$, $k = 1, 2, \dots, 9$. Ο επαναληπτικός αυτός αλγόριθμος πολλαπλασιασμού είναι μια παραλλαγή του αλγορίθμου που επινόησε ο Al Khwarizmi και που χρησιμοποιείται σήμερα σε αρκετές Ευρωπαϊκές χώρες [3].

Όσον αφορά τον τρόπο επικοινωνίας με τους δύο μετατροπείς, MCP3202 και MCP4821, αυτός έχει αναλυθεί διεξοδικά στις υποενότητες 5.3.1 και 5.3.2 αντίστοιχα. Το μόνο που πρέπει να προσθέσουμε είναι πως προκειμένου το Cog0 να παράγει τις συχνότητες ρολογιού που χρειάζονται οι δύο μετατροπείς, αξιοποιεί τον ένα ενσωματωμένο μετρητή του, COUNTERA. Ο μετρητής αυτός προγραμματίζεται για “Διαφορική PLL” λειτουργία, παράγοντας στους ακροδέκτες P16 και P20 του Propeller, παλμούς συχνότητας 625 KHz. Όπως βλέπουμε από το ηλεκτρονικό διάγραμμα 5.19 ο ακροδέκτης P16 συνδέεται με την είσοδο SCK του DAC, ενώ ο P20 με την είσοδο CLK του ADC. Τα υπόλοιπα σήματα που απαιτούνται για την επικοινωνία με τους δύο

⁵Ο κώδικας αυτός της σελίδας 380 του Propeller Manual, αφορά τον πολλαπλασιασμό μεταξύ δύο μη προσημασμένων ακεραίων, μήκους 16 bits ο καθένας.

Ψευδοκώδικας 3 Αλγόριθμος πολλαπλασιασμού μεταξύ ενός προσημασμένου ακεραίου X(16 bits) και ενός μη προσημασμένου ακεραίου Y(9 bits)

Require: X: 16 bits Signed , Y: 9 bits Unsigned

Ensure: Result in X

$X_{abs} \leftarrow ABS(X)$ {Βρες την απόλυτη τιμή της προσημασμένης μεταβλητής X}

$Y^{[0]} = Y$

for $k = 1$ to 9 do

$$Y^{[k]} = \begin{cases} Y^{[k-1]}/2 & \text{εάν } Y^{[k-1]} \text{ άρτιος,} \\ Y^{[k-1]}/2 + 2^8 X_{abs} & \text{εάν } Y^{[k-1]} \text{ περιττός} \end{cases}$$

end for

$Y \leftarrow Y^{[9]}$ {Αποθήκευσε το αποτέλεσμα του πολλαπλασιασμού, $Y^{[9]}$, στην μεταβλητή Y}

if $X < 0$ then {Ανάκτηση προσήμου}

$X \leftarrow (-Y)$

else

$X \leftarrow Y$

end if

Χρόνος εκτέλεσης ενός βρόχου για το Cog0	
ADC Section	28,8 μsec
DAC Section	25,6 μsec
Gamma Section	3,1 μsec
Various config	0,2 μsec
Sum (T_s)	57,7 μsec
f_s	17331 Hz

Πίνακας 5.4: Υπολογισμός συχνότητας δειγματοληψίας f_s

μετατροπείς - χρησιμοποιώντας το πρωτόκολλο SPI - παράγονται από τον κώδικα που αναπτύξαμε.

Από την δομή του κώδικα που εκτελείται στο Cog0 (βλ. σχήμα 5.20), είναι φανερό πως η περίοδος δειγματοληψίας T_s , ισούται με τον χρόνο που διαρκεί ένα loop ($ADC\ section + ECHO\ section + DAC\ section$). Στον πίνακα 5.4 μπορούμε να δούμε τον συνολικό χρόνο εκτέλεσης ενός βρόχου, καθώς και το πώς ο χρόνος αυτός κατανομείται σε κάθε section. Βλέπουμε λοιπόν πως η συχνότητα δειγματοληψίας του συστήματός μας είναι περίπου 17 ΚHz. Τέλος να σημειώσουμε πως η ψευδοεπιλογή $old \leftarrow old \times feedback/512$ διαρκεί 1,85 μsec, δηλαδή το 60% του συνολικού χρόνου του Echo Section.

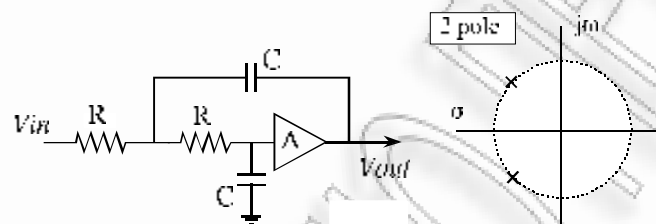
5.4 Ανάλυση του Αναλογικού υποσυστήματος

Το κυριότερο κομμάτι του αναλογικού υποσυστήματος (σχήμα 5.4), είναι τα δύο πανομοιότυπα βαθυπερατά φίλτρα, ADC filter και DAC filter, τα οποία φιλτράρουν τα αναλογικά σήματα που σχετίζονται με τους δύο μετατροπείς.

Ως γνωστόν προτού τροφοδοτήσουμε έναν A-D μετατροπέα, πρέπει να εξαλείψουμε όλες τις συχνότητες του σήματος εισόδου που είναι μεγαλύτερες από $f_s/2$, όπου f_s είναι η συχνότητα δειγματοληψίας. Αν δεν συμβεί κάτι τέτοιο, τότε κατά την μετατροπή του ψηφιακού σήματος σε αναλογικό θα παρουσιαστούν συχνότητες μέσα στο διάστημα $[0, f_s/2]$ που δεν υπήρχαν στο αρχικό μας σήμα, με αποτέλεσμα την αλλοίωσή του. Σύμφωνα με τις αρχικές προδιαγραφές του συστήματός μας (βλ. πίνακα 4.1), η συχνότητα δειγματοληψίας είναι ίση προς, $f_s = 20\ KHz$ και η μεγαλύτερη συχνότητα του σήματος εισόδου είναι, $f_{max} = 5\ KHz$. Έτσι προκειμένου να εξαλείψουμε το συχνотικό περιεχόμενο του σήματος εισόδου από τα 10 ΚHz και πάνω, σχεδιάσαμε και υλοποιήσαμε ένα βαθυπερατό φίλτρο Butterworth τέταρτης τάξης, με συχνότητα αποκοπής στα 5 ΚHz, όπως βλέπουμε στον πίνακα 5.5. Το ίδιο το φίλτρο απλό είναι τέταρτης τάξης, ο βαθμός εξαθροίνισης του είναι 21 dB/octave, με αποτέλεσμα στα 10 ΚHz το πλάτος του σήματός μας να έχει φτάσει στο

Τεχνικά χαρακτηριστικά για τα δύο βαθυπερατά φίλτρα	
Τύπος φίλτρου	Butterworth (flattest amplitude)
Τάξη φίλτρου	4
Συχνότητα αποκοπής	5 KHz
Βαθμός εξασθένησης	24 dB/οκταβά
Κέρδος φίλτρου	$K = 2.5$
Τοπολογία	Sallen-Key, Equal component value (2 στάδια)

Πίνακας 5.5: Τεχνικά χαρακτηριστικά για τα δύο βαθυπερατά φίλτρα (ADC, DAC)

Σχήμα 5.21: Τοπολογία Sallen-Key. Το κύκλωμα αυτό παράγει δύο πόλους οι οποίοι βρίσκονται σε έναν κύκλο με ακτίνα $1/RC$ στο s -επίπεδο

6.3%¹⁶ της αρχικής του τιμής.

Η βασική δομή για ένα δεύτερης τάξης φίλτρο με τοπολογία "Sallen-Key, equal component value" φαίνεται στο σχήμα 5.21 [12]. Βλέπουμε πως οι δύο αντιστάσεις έχουν την ίδια τιμή όπως επίσης και οι δύο πυκνωτές. Αυτό έχει ως συνέπεια οι δύο πόλοι του φίλτρου να βρίσκονται σε κύκλο ακτίνας $1/RC$ στο επίπεδο s . Η θέση τους πάνω στον κύκλο - η οποία καθορίζει το πόσο "απότομο" είναι το φίλτρο - εξαρτάται από το κέρδος του φίλτρου, το οποίο κυμαίνεται από 1 μέχρι 3. Όσο πιο κοντά βρίσκονται οι πόλοι αυτοί στον μιγαδικό άξονα, τόσο πιο μεγάλο το κέρδος του φίλτρου και τόσο μεγαλύτερη είναι η τάση του για τετραλάντωση [12]. Χρησιμοποιώντας τη κίνηση Sallen-Key, μπορούμε να υλοποιήσουμε πολλών ειδών φίλτρα μεταξύ των οποίων βρίσκεται και το φίλτρο Butterworth που χρησιμοποιήσαμε.

Το δικό μας φίλτρο - όπως βλέπουμε στο σχήμα 5.22 - αποτελείται από δύο στάδια κάθε ένα από τα οποία ακολουθεί την τοπολογία του σχήματος 5.21. Οι τιμές των εξαρτημάτων $R = 10\text{ K}\Omega$, $C = 3.3\text{ nF}$ είναι ίδιες και για τα δύο στάδια, με αποτέλεσμα να έχουμε συνολικά 4 πόλους - 2 για κάθε στάδιο - που βρίσκονται επί του ιδίου κύκλου, με συχνό, ημι σφαιρικής, $f_c = 1/2\pi RC = 4823\text{ Hz}$. Το κέρδος του κάθε σταδίου είναι διαφορετικό, όπως για το πρώτο στάδιο ισχύει $K_1 = 1 + 5.6/39 = 1.144$, ενώ για το δεύτερο, $K_2 = 1 + 47/39 = 2.205$. Έτσι το συνολικό κέρδος του φίλτρου θα είναι, $K = K_1 \times K_2 = 2.52$. Ο συνδυασμός αυτός είναι που δίνει στο φίλτρο μας την ιδιότητα να έχει την μεγαλύτερη δυνατή απότομη μετάβαση, χωρίς να υπάρχει κυμάτωση στην περιοχή της ζώνης διέλευσης (passband frequencies)¹⁷. Στο σημείο αυτό πρέπει να σημειώσουμε πως για την σχεδίαση και την υλοποίηση του συγκεκριμένου φίλτρου, χρησιμοποιήσαμε τον οδηγό που βρίσκεται στο έβδομο βιβλίο του Don Lancaster, "Active Filter Cookbook" (σελίδα 112, αναφορά [4]).

Προκειμένου να τροφοδοτήσουμε τον A-D μετατροπέα με το φιλτραρισμένο σήμα από το antialiasing φίλτρο, θα πρέπει να το ενισχύσουμε κατάλληλα έτσι ώστε το πλάτος του να καλύπτει το διάστημα από 0 έως 5 Volts, αφού η τάση αναφοράς του μετατροπέα είναι $V_{ref} = 5\text{ Volts}$. Εδώ πρέπει να σημειώσουμε πως το σήμα εισόδου για το σύστημά μας, είναι αυτό μιας τυπικής ηλεκτρικής κιθάρας, με αποτέλεσμα το εύρος του να είναι, $V_{in} = 0.5\text{ V}_{p-p}$. Εάν $K = 2.5$ είναι το κέρδος του antialiasing φίλτρου και G , η ζητούμενη ενίσχυση, τότε προφανώς ισχύει η σχέση:

$$(5 - 0) = K \times G \times 0.5 \rightarrow G = \frac{5}{2.5 \times 0.5} \rightarrow G = 4$$

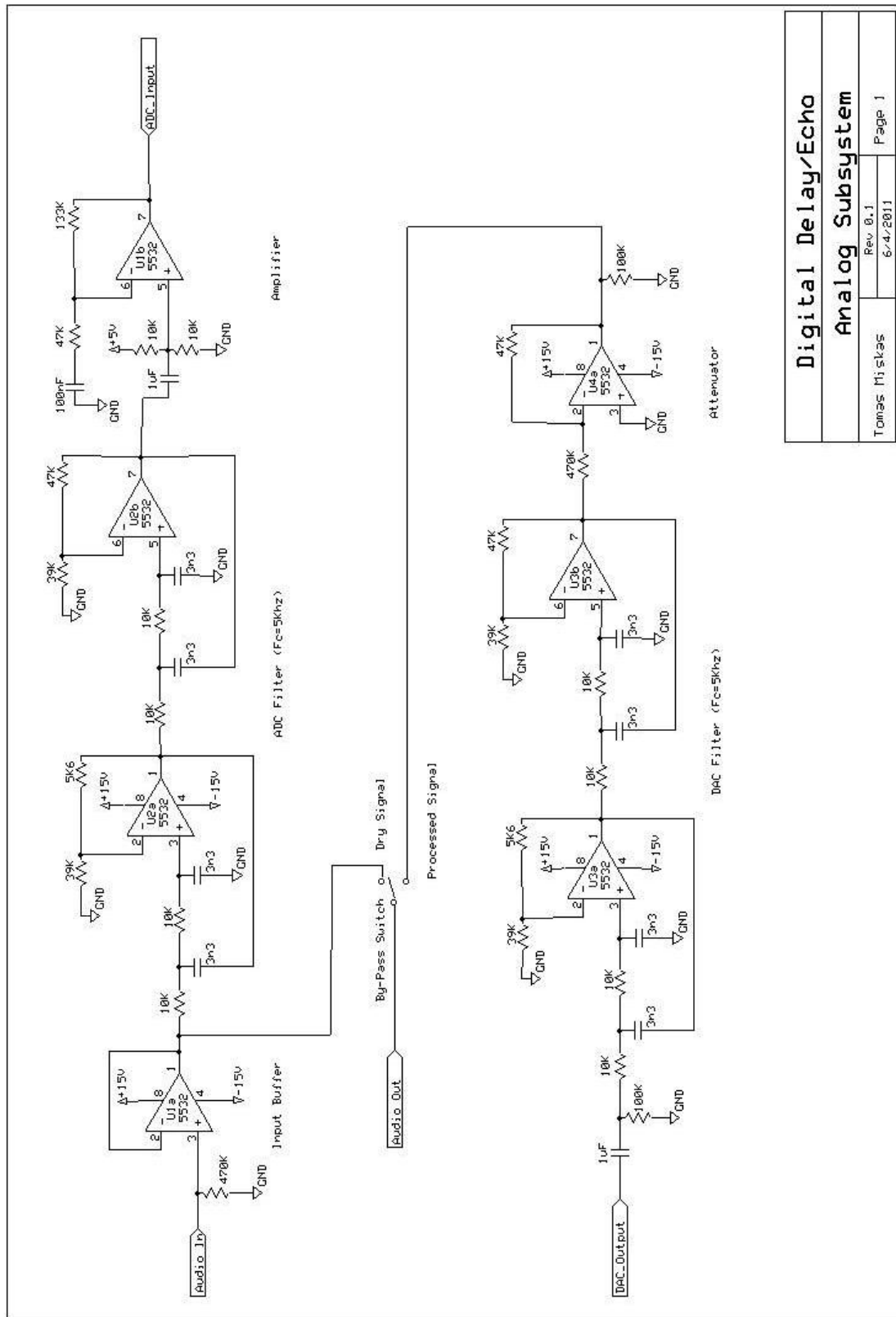
¹⁶ 24 dB/οκταβά σημαίνει ότι σε κάθε δεκάπλασμα της συχνότητας αποκοπής, θα έχουμε εξασθένηση κατά 24 dB

¹⁷ Ένα φίλτρο με την ιδιότητα αυτή ονομάζεται φίλτρο Butterworth.

Όπως φαίνεται στο ηλεκτρονικό διάγραμμα του σχήματος 5.22, το τμήμα του αναλογικού υποσυστήματος που αφορά τον A-D μετατροπέα, υλοποιείται από τους δύο (διπλούς) τελεστικούς ενισχυτές, U1, U2. Το anti-aliasing φίλτρο αφορά τον ένα διπλό τελεστικό U2 (U2a, U2b), ενώ η απαιτούμενη προαναφερόμενη ενίσχυση της εξόδου του φίλτρου κατά 4 φορές, πραγματοποιείται από το U1b.

Όσον αφορά το τμήμα που έχει σχέση με τον D-A μετατροπέα, η έξοδος του DAC συνδέεται με ένα βαθυπερατό φίλτρο πανομοιότυπο με το anti-aliasing φίλτρο που περιγράψαμε. Το φίλτρο αυτό υλοποιείται από τον διπλό τελεστικό U3 (U3a, U3b). Στην συνέχεια το φιλτραρισμένο σήμα διέρχεται μέσω του U1a όπου εξασθενεί κατά 10 φορές, με αποτέλεσμα η στάθμη του να γίνει ίδια με αυτήν του σήματος εισόδου, δηλαδή $V_{out} = 0.5 V_{p-p}$. Πράγματι, το αναλογικό σήμα που παίρνουμε από το DAC κυμαίνεται από 0 – 2.048 Volts. Το κέρδος του DAC φίλτρου είναι περίπου ίσο με $K = 2.5$, με αποτέλεσμα το σήμα μετά την έξοδό του από αυτό να παίρνει τιμές από 0 – 5 Volts. Άρα μια εξασθένηση κατά 10 φορές έχει ως αποτέλεσμα το επεξεργασμένο σήμα μας να έχει πλάτος περίπου ίδιο με αυτό του αρχικού εισόδου.

Τέλος προκειμένου να μπορέσουμε να αξιολογήσουμε την επίδραση του συστήματος που υλοποιήσαμε στο σήμα εισόδου, προσθέσαμε έναν διακόπτη by pass μέσω του οποίου μπορούμε να έχουμε στη έξοδό μας είτε το αυθεντικό σήμα εισόδου (dry signal), είτε το επεξεργασμένο (processed signal), όπως βλέπουμε στο ηλεκτρονικό διάγραμμα του σχήματος 5.22.



Digital Delay/Echo	
Analog Subsystem	
Tomas Hiskas	Rev. 0.1
	6/4/2011
	Page 1

Σχήμα 5.22: Ηλεκτρονικό κύκλωμα του αναλογικού τμήματος ήχου

Κεφάλαιο 6

Συμπεράσματα-Περίληψη

Στον πίνακα 6.1 μπορούμε να δούμε τις τελικές προδιαγραφές του συστήματος που υλοποιήσαμε, οι οποίες δεν διαφέρουν ιδιαίτερα με αυτές που είχαμε θέσει εξ αρχής (βλ. πίνακα 4.1). Ο μέγιστος χρόνος καθυστέρησης είναι γύρω στα 700 μ s και η συχνότητα δειγματοληψίας περίπου 17 KHz. Να επενθούμε πως ο μέγιστος χρόνος καθυστέρησης εξαρτάται από την χρησιμοποιούμενη μνήμη RAM - που στην περίπτωση μας είναι 12320 words - και από την συχνότητα δειγματοληψίας.¹

Όπως είδαμε στο προηγούμενο κεφάλαιο χρησιμοποιήθηκαν μόνο δύο εκ των οκτώ διαθέσιμων επεξεργαστών του Propeller chip, αξιοποιώντας το γεγονός της υψηλής συχνότητας λειτουργίας του ρολογιού του στα 89 MHz. Έτσι μόνο ένας επεξεργαστής (Core0) έφερε εις πέρας τις τρεις απαιτούμενες εργασίες (Επικοινωνία με το ADC, Αλγόριθμος Echo, Επικοινωνία με το DAC) σειριακά - δηλαδή την μία μετά την άλλη - χωρίς μεγάλη επιβάρυνση στον συνολικό χρόνο διάρκειας ενός κύκλου του επαναληπτικού βρόχου. Αποτέλεσμα αυτού του γεγονότος είναι να έχουμε μια πολύ ικανοποιητική συχνότητα δειγματοληψίας για το δεδομένο συχνοτικό εύρος του σήματος εισόδου. Πράγματι η συχνότητα δειγματοληψίας είναι πάνω από τρεις φορές μεγαλύτερη της μέγιστης συχνότητας που μπορεί να έχει το σήμα εισόδου.

Προκειμένου να μπορέσει ο αναγνώστης να αξιολογήσει το σύστημά μας, ηχογραφήσαμε μερικά ηχητικά δείγματα του σήματος εισόδου του, υπό την μορφή των αρχείων (βλ. συνοδευτικό CD). Το σήμα εισόδου είναι αυτό μιας ηλεκτρικής κιθάρας. Αρχικά στο αρχείο "*dry.wav*" έχουμε ηχογραφήσει ένα δείγμα όπου έχουμε παρακάμψει την επίδραση του εφέ μέσω του διακόπτη By-Pass, με αποτέλεσμα ο ήχος να μην έχει υποστεί καμία επεξεργασία. Η ύπαρξη των δύο ποτενοσιμέτρων ελέγχου - που καθορίζουν τον χρόνο καθυστέρησης και το κέρδος της ανάδρασης - δίνει την δυνατότητα στον χρήστη να πειραματιστεί, δημιουργώντας ενδιαφέροντα μουσικά εφέ όπως συμβαίνει με ανάλογα εμπορικά προϊόντα τύπου Digital Delay. Έτσι τα υπόλοιπα ηχογραφημένα δείγματα κφορούν διάφορους συνδυασμούς τιμών των δύο αυτών παραμέτρων, φωτίζοντας ένα μέρος των δυνατοτήτων του συστήματος.

Το σύστημα που κτίσαμε κέρφασε μπορεί να δώσει χοντρικά την αίσθηση της ακουστικής συμπεριφοράς ενός συνηθισμένου δωμάτιου, αν θέσουμε τον χρόνο καθυστέρησης γύρω στα 60 μ s. Από τους ηχογραφημένους ήχους "*rev1_lowfb.wav*", "*rev2_medfb.wav*" και "*rev3_maxfb.wav*" μπορούμε να αντιληφθούμε την επίδραση του κέρδους της ανάδρασης στην περίπτωση αυτή. Έτσι στο πρώτο αρχείο το κέρδος είναι χαμηλό ενώ στο τρίτο έχει την μεγαλύτερη δυνατή τιμή (0.57) που επιτρέπει το σύστημα. Όπως έχουμε ήδη αναφέρει, το feedback coil² φίλτρο που υλοποιήθηκε, στην πραγματικότητα εξομοιώνει έναν κυλινδρικό σωλήνα με τα δύο του άκρα είτε ανοικτά είτε κλειστά. Έτσι για τις τιμές των παραμέτρων που αναφέραμε, μπορούμε να πούμε πως προσεγγίζουμε την συμπεριφορά των ακουστικών κυμάτων καθώς αυτά διαδίδονται σε έναν χώρο που περιόριζεται μόνο από δύο τοίχους.

Όσον αφορά την ποιότητα του παραγόμενου ήχου, αυτή κρίνεται πολύ ικανοποιητική αν λάβουμε υπόψιν ότι δεν υπάρχει επίπεδο γης (ground plane), αφού όλη η κατασκευή έγινε σε veroboard. Έτσι η υψηλότερη συχνότητα λειτουργίας του ψηφιακού υποσυστήματος σίγουρα επηρεάζει το πολύ ευαίσθητο σε τέτοιες παρεμβολές, ηχητικό σήμα. Να σημειώσουμε επίσης πως οι δύο μετατροπείς που χρησιμοποιήθηκαν (ADC, DAC), δεν προσδίδονται για ακουστικά σήματα, σύμφωνα με τα φύλλα οδηγιών τους. Παρόλα αυτά, η (αναγκαστική) επιλογή τους μας δικαίωσε με το παραπάνω, αποδεικνύοντας πως η χαμηλή ανάλυση των 12 bits είναι επαρκής για σήματα ηλεκτρικών μουσικών οργάνων.

Με δεδομένο ότι αξιοποιήσαμε μόνον το 1/4 της υπολογιστικής ισχύος του Propeller chip, καταλαβαίνει κανείς ότι με αφετηρία αυτό το απλό σύστημα μπορούν να υλοποιηθούν και άλλα ηχητικά εφέ, όπως *flanger*, *chorus*, *reverb*, *pitch shifter*, με δυνατότητα επιλογής των, είτε μεμονωμένα, είτε με κάποιο συνδυασμό (πχ *chorus* + *reverb*). Τα εφέ αυτά είναι δυνατόν να πραγματοποιηθούν, σφαιρί η λειτουργία τους στηρίζεται σε

¹ Προφανώς ισχύει $Delay_{max} = 12320/17331 = 0.71sec$.

Τελικές Προδιαγραφές του συστήματος Digital Delay	
Μικροελεγκτής	Propeller chip
Μνήμη	32 Kbytes static Ram ενσωματωμένη στο Propeller chip
A-D Μετατροπείς	MCP3202 (Successive Approximation, 12 bit)
D-A Μετατροπείς	MCP1821 (Resistive String, 12 bit)
Τύπος anti-aliasing φίλτρου	Butterworth τέταρτης τάξης με $f_c = 5 \text{ KHz}$
Τύπος DAC φίλτρου	Butterworth τέταρτης τάξης με $f_c = 5 \text{ KHz}$
Συχνότητα δειγματοληψίας	17331 Hz
Ανάλυση	12 bits
Delay time (ms)	30 – 711
Feedback gain	0.03 – 0.57
Απόκριση Συχνότητων	20 Hz – 5 KHz
Τύση τροφοδοσίας	+/- 15 Volls

Πίνακας 6.1: Τελικές προδιαγραφές του συστήματος

μια γραμμή καθυστέρησης όπως και το εφέ που κατασκευάσαμε.

Κλείνοντας την εργασία αυτή, μπορούμε να συμπεράνουμε πως τελικά είναι δυνατή η ψηφιακή ολοποίηση απλών ηχητικών εφέ που βασίζονται σε λίγες αριθμητικές πράξεις, ακόμα και αν δεν διαθέτουμε εξειδικευμένο επεξεργαστή σήματος (DSP) με ενσωματωμένη αριθμητική παγίδα κινητής υποδιαστολής (FPU). Βέβαια το τελικό αποτέλεσμα σίγουρα δεν είναι το ίδιο, αλλά όταν έχουμε να κάνουμε με χαμηλής πιστότητας ακουστικά σήματα (όπως αυτό της ηλεκτρικής κιθάρας), μια χαμηλού κόστους προσπάθεια όπως είναι η δική μας μπορεί να έχει αποδεκτά αποτελέσματα.

Βιβλιογραφία

- [1] Παπαχρήστου Ανδρέας, *Περί Μουσικής και Ηχοηφίας*, Μουσικός Οίκος ΦΩάνης Νάνας, 2001.
- [2] Tom Borner and Ali Yalgin, *Design of Industrial Information Systems*, Academic Press, 2006.
- [3] S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani., *Algorithms*, McGraw Hill, 2006.
- [4] Don Lancaster, *Active-Filter Cookbook*, Howard W. Sams & Co. Inc. first edition, 1985.
- [5] Andy Lindsay, *Propeller Education Kit Labs: Fundamentals Version 1.2*, Parallax inc. <http://www.parallax.com/Portals/0/Downloads/docs/prod/prop/PEKitLabs-v1.2.pdf>, 2006.
- [6] Microchip Technology Inc, <http://ww1.microchip.com/downloads/en/DeviceDoc/21034e.pdf>. MCP3202 Datasheet: 2.7V Dual Channel 12bit A/D Converter with SPI Serial Interface, 2006.
- [7] Microchip Technology Inc, <http://ww1.microchip.com/downloads/en/DeviceDoc/22274B.pdf>. MCP4801/4811/4821 Datasheet: 8/10/12-Bit Voltage Output Digital to Analog Converter with Internal V_{ref} and SPI Interface, 2010.
- [8] J. A. Moorer, About this reverberation business, *Computer Music Journal*, 3(2):13–28, 1979.
- [9] Tom O'Haver, Audio processing with a microprocessor, adding a virtual tape loop, *Byte Magazine*, June 1978.
- [10] Robert Pepp, Sound effects processing, *Circuit Cellar*, (216), July 2008.
- [11] Parallax Inc, <http://www.parallax.com/Portals/0/Downloads/docs/prod/prop/webPM-v1.1.pdf>, *Propeller Manual V1.1*.
- [12] Steven Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, second edition, 1999.
- [13] Ker Steiglitz, *A DSP Primer*, Prentice-Hall, Englewood Cliffs, NJ, 1996.
- [14] Udo Zolzer, editor, *Digital Audio Effects*, John Wiley & Sons, Inc., New York, first edition, 2002.