

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Η/Υ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

*ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΔΙΑΜΟΡΦΩΣΗΣ ΕΥΡΟΥΣ ΠΑΛΜΩΝ ΜΕ
ΧΡΗΣΗ ΕΠΕΞΕΡΓΑΣΤΗ DSP.*

ΜΟΙΡΟΓΙΩΡΓΟΥ ΚΩΝΣΤΑΝΤΙΑ

Εξεταστική Επιτροπή:

Καθ. Καλαϊτζάκης Κων/νος (επιβλέπων)
Καθ. Σταυρακάκης Γιώργος
Διδάσκων ΠΔ407 Κουτρούλης Ευτύχης

Χανιά 2003

Στην οικογένειά μου
Στο Διονύση

Ευχαριστίες

Στον Καθηγητή Καλαϊτζάκη Κων/νο για την ευκαιρία που μου έδωσε να ασχοληθώ και να εξειδικεύσω τις γνώσεις μου στον τομέα του υλικού και των DSPs ειδικότερα. Επίσης, θα ήθελα να τον ευχαριστήσω για την ανοχή και τη βοήθειά του μέχρι το πέρας της εργασίας.

Στον Καθηγητή Σταυρακάκη Γιώργο για τη συνεισφορά του ως μέλος της εξεταστικής επιτροπής.

Στον Διδάσκων ΠΔ407 Κουτρούλη Ευτύχη για τη βοήθεια, επίβλεψη και αδιάκοπη υποστήριξη της προσπάθειάς μου.

Στους φίλους μου.

Στο Διονύση για τις πολύτιμες γνώσεις του και την προσοχή που μου έδειξε...

Στο Διονύση γιατί πίστεψε σε μένα όσο, ίσως, κανείς...

Στο Διονύση για την αγάπη του...

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια ο τομέας των Ηλεκτρονικών Ισχύος έχει παρουσιάσει ιδιαίτερη ανάπτυξη. Τα συστήματα Ηλεκτρονικών Ισχύος αποτελούνται από έναν ή περισσότερους μετατροπείς ηλεκτρονικών ισχύος, οι οποίοι χρησιμοποιούν ημιαγώγιμα στοιχεία ισχύος ελεγχόμενα από ολοκληρωμένα κυκλώματα. Ο μετατροπέας είναι το βασικό στοιχείο αυτών των συστημάτων. Ελέγχει και μορφοποιεί μία είσοδο πλάτους V_i , συχνότητας f_i και αριθμού φάσεων m_i σε έξοδο πλάτους V_o , συχνότητας f_o και αριθμού φάσεων m_o . Σε DC-AC μετατροπή, ο μετατροπέας ονομάζεται αντιστροφέας (inverter).

Στην παρούσα διπλωματική εργασία, μελετήθηκε ένας PWM Inverter, ένας αντιστροφέας με έξοδο σήμα ελεγχόμενου πλάτους και συχνότητας. Με στόχο η τάση εξόδου του να μοιάζει όσο περισσότερο γίνεται με ημίτονο, εφαρμόστηκαν μέθοδοι Διαμόρφωσης Εύρους Παλμών. Πιο συγκεκριμένα, υλοποιήθηκε η μέθοδος Διαμόρφωσης Εύρους Παλμών με Μονοπολική Τάση Εξόδου (PWM with Unipolar Voltage Switching) και Διαμόρφωση Εύρους Παλμών με Διπολική Τάση Εξόδου (PWM with Bipolar Voltage Switching), ενώ έγινε μελέτη του τρόπου υλοποίησης της μεθόδου Three-Phase PWM.

Για την υλοποίηση των παραπάνω έγινε χρήση του DSP επεξεργαστή TMS320C31 της Texas Instruments. Οι DSP επεξεργαστές χρησιμοποιούνται, κυρίως, σε συστήματα επεξεργασίας σήματος και τα πλεονεκτήματά τους είναι πολλά. Η ανάπτυξη του κώδικα έγινε σε γλώσσα προγραμματισμού assembly.

ΠΕΡΙΕΧΟΜΕΝΑ:

ΕΙΣΑΓΩΓΗ	8
ΚΕΦΑΛΑΙΟ 1^ο : ΕΙΣΑΓΩΓΗ ΣΤΗΝ TMS320C3X ΣΕΙΡΑ DSPS.....	10
1. Εισαγωγή.....	10
2. Αρχιτεκτονική των 'C3x DSPs.	11
2.1 Central Processing Unit (CPU)	12
2.1.1 Πολλαπλασιαστής ακέραιων/δεκαδικών αριθμών.....	13
2.1.2 Αριθμητική και λογική μονάδα και εσωτερικά buses(διάυλοι).....	13
2.1.3 Βοηθητικές αριθμητικές μονάδες καταχωρητών (ARAs).....	13
2.2 Πρωτεύον αρχείο καταχωρητών της CPU (CPU Primary Register File).....	13
2.3 Άλλοι καταχωρητές.....	17
2.4 Οργάνωση μνήμης.....	17
2.5 Λειτουργία εσωτερικού bus.....	18
2.6 Λειτουργία εξωτερικού bus.....	19
2.7 Περιφερειακά.....	20
2.7.1 Timers.....	20
2.7.2 Σειριακή Θύρα.....	23
3. Διευθυνσιοδότηση (Addressing).....	23
3.1 Κυκλική διευθυνσιοδότηση (Circular Buffering).....	24
4. Interrupts.....	26
5. Εντολές της γλώσσας Assembly.....	28
5.1 Load και Store εντολές.....	28
5.2 Εντολές δύο τελεστών.....	28
5.3 Εντολές τριών τελεστών.....	28
5.4 Εντολές ελέγχου του προγράμματος (program - control instructions).....	29
5.5 Παράλληλες εντολές.....	29
6. Pipeline.....	29
6.1 Η δομή του pipeline.....	29
6.2 Τα pipeline conflicts.....	31
7. Εφαρμογές της TMS320 οικογένειας επεξεργαστών.....	31
8. Πλεονεκτήματα των DSPs.....	32
ΚΕΦΑΛΑΙΟ 2^ο : ΗΛΕΚΤΡΟΝΙΚΑ ΙΣΧΥΟΣ – SPWM.....	34
1. Εισαγωγή.....	34
2. Κατηγορίες μετατροπέων (converters).....	35
3. DC –AC Αντιστροφείς: dc \Leftrightarrow Ημιτονοειδές ac.....	36
4. Αντιστροφείς διαμόρφωσης εύρους παλμών.....	37
4.1. Αντιστροφείς πλήρους γέφυρας (Full-bridge inverters).....	40
4.1.1. Διαμόρφωση PWM με διπολική τάση εξόδου (Bipolar Voltage Switching).....	41
4.1.2. Διαμόρφωση PWM με μονοπολική τάση εξόδου (Unipolar Voltage Switching).....	42
5. Ημιτονοειδής διαμόρφωση εύρους παλμών (SPWM) με χρήση επεξεργαστή DSP (TMS320C31).....	45
5.1 Εισαγωγή.....	45
5.1.1 Γενική περιγραφή υλοποίησης της SPWM στον 'C31.....	46

5.1.2 Εναλλακτικός τρόπος υπολογισμού του Duty Cycle της μεθόδου.	46
ΚΕΦΑΛΑΙΟ 3^ο : ΥΛΟΠΟΙΗΣΗ ΤΟΥ SPWM ΣΤΟΝ TMS320C31.	48
1. Εισαγωγή	48
2. Ανάλυση κώδικα	49
3. Παρατηρήσεις	56
4. Υλοποίηση της PWM με διπολική τάση εξόδου (PWM with Bipolar Voltage Switching).....	57
5. Επέκταση του κώδικα για υλοποίηση three – phase SPWM.	59
ΚΕΦΑΛΑΙΟ 4^ο : ΠΡΟΣΟΜΟΙΩΣΗ ΚΥΚΛΩΜΑΤΟΣ ΙΣΧΥΟΣ - ΑΠΟΤΕΛΕΣΜΑΤΑ	61
1. Προσομοίωση Κυκλώματος Ισχύος.....	61
1.1 PWM με μονοπολική τάση εξόδου.....	61
1.2 PWM με μονοπολική τάση εξόδου.....	64
2. Πειραματικά αποτελέσματα – Μετρήσεις	65
2.1 PWM με μονοπολική τάση εξόδου	65
2.2 PWM με διπολική τάση εξόδου	72
ΚΕΦΑΛΑΙΟ 5^ο : ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	75
1. Συμπεράσματα	75
2. Μελλοντικές επεκτάσεις.....	75
ΠΑΡΑΡΤΗΜΑ Α: TMS320C3X DSP STARTER KIT.....	77
1. Εισαγωγή	77
2. Εγκατάσταση του DSK Assembler και Debugger.....	78
3. Σύντομη περιγραφή της ανάπτυξης κώδικα στο DSK.....	82
3.1 Περιγραφή του DSK Assembler.....	82
3.2 Περιγραφή του DSK Debugger	83
4. Χρησιμοποιώντας τον DSK Assembler.....	84
5. Χρησιμοποιώντας τον DSK Debugger.	85
Α) Παράθυρο Disassembly.....	86
Β) Παράθυρο των CPU REGISTER.....	86
Γ) Παράθυρο Μνήμης (Memory Window)	87
Δ) Παράθυρο εντολών (Command Window)	88
6. Εντολές του Debugger.....	88
7. Το TLC32040 AIC.	91
ΒΙΒΛΙΟΓΡΑΦΙΑ	92

ΣΧΗΜΑΤΑ:

Σχ.1.1: Block diagram των επεξεργαστών TMS320C3x	11
Σχ.1.2: Κεντρική Μονάδα Επεξεργασίας (CPU).....	12
Σχ.1.3: Διάταξη καταχωρητή εκτεταμένης ακρίβειας	14
Σχ.1.4: Interrupt Enable Register (IE)	15
Σχ.1.5: I/O-Flag Register (IOF).....	16
Σχ.1.6: Memory map του TMS320C31	18
Σχ.1.7: Εσωτερικό busing του TMS320C31	19
Σχ.1.8: Peripheral bus memory-mapped registers	21
Σχ.1.9: Θέσεις μνήμης των interrupts.....	22
Σχ.1.10: Παραγωγή των index και top, bottom του κυκλικού buffer.....	24
Σχ.1.11: Τα τέσσερα επίπεδα του pipeline	30
Σχ. 2.1: Αντιστροφέας τύπου γέφυρας (bridge-type inverter).....	36
Σχ.2.2: One-leg switch-mode inverter	38
Σχ.2.3: Διαμόρφωση εύρους παλμών	38
Σχ.2.5: Ημιτονοειδές PWM	40
Σχ.2.6: Μονοφασικός αντιστροφέας πλήρους γέφυρας	41
Σχ.2.7: PWM με διπολική τάση εξόδου	42
Σχ.2.8: Φάσμα αρμονικών	42
Σχ. 2.9: PWM με μονοπολική τάση εξόδου	43
Σχ.2.10: Επιμέρους τάσεις εξόδου του inverter	44
Σχ.2.11: Φάσμα αρμονικών	45
Σχ.2.12: Έξοδος του inverter	45
Σχ.2.13: Παραγωγή του PWM σήματος σε μία περίοδο μετάβασης	47
Σχ.3.1 Flow Chart του αλγόριθμου υλοποίησης της μεθόδου PWM με μονοπολική τάση εξόδου	50
Σχ.3.2 Συσχετισμός των δύο βασικών σημάτων της μεθόδου PWM για μία περίοδο ημιτόνου	53
Σχ.3.3 Συσχετισμός των δύο Timers και χρόνοι μεταξύ διαδοχικών INTS	54
Σχ.3.4 Flow Chart του αλγόριθμου υλοποίησης της μεθόδου PWM με διπολική τάση εξόδου	58
Σχ.3.5 Flow Chart του αλγόριθμου υλοποίησης της μεθόδου Three-Phase PWM	59

Σχ.4.1 Κύκλωμα αναστροφής του σήματος V_2	62
Σχ.4.2 Κύκλωμα αθροιστή	62
Σχ.4.3 Κύκλωμα χαμηλοπερατού φίλτρου δύο πόλων.....	63
Σχ.4.4α) Σήμα εξόδου V_1 του 'C31, β) Σήμα εξόδου V_2 του 'C31,	66
γ) Σήμα εισόδου του φίλτρου, δ) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου).....	66
Σχ.4.5α) Σήμα εξόδου V_1 του 'C31, β) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου) .	67
Σχ.4.6α) Σήμα εξόδου V_1 του 'C31, β) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου) .	68
Σχ.4.7α) Σήμα εξόδου V_1 του 'C31, β) Σήμα εξόδου V_2 του 'C31, γ) Σήμα εισόδου του φίλτρου, δ) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου).....	69
Σχ.4.8α) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)	70
Σχ.4.9α) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)	71
Σχ.4.10α) Σήμα εξόδου V_1 του 'C31, β) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)	72
Σχ.4.11α) Σήμα εξόδου V_1 του 'C31, β) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)	73
Σχ.4.12α) Σήμα εξόδου V_1 του 'C31, β) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)	74
Σχ.Α.1 TMS320C3x DSK Block Diagram.....	78
Σχ.Α.2 Σύνδεση του DSK με το PC και την παροχή ρεύματος.....	79
Σχ.Α.3 Βασική οθόνη debugger	80
Σχ.Α.4 Κακή εγκατάσταση debugger.....	81
Σχ.Α.5 Ανάπτυξη κώδικα στο DSK	82
Σχ.Α.6 Βασική οθόνη του Debugger	83
Σχ.Α.7 Παράθυρο Disassembly του DSK debugger	86
Σχ.Α.10 Command Window του debugger	88

ΠΙΝΑΚΕΣ:

Πίνακας 1.1: Περίληψη των bits του IOF Register	16
Πίνακας 1.2: Διευθύνσεις των Interrupt service routines του 'C31	26
Πίνακας 1.2: Προτεραιότητες των Interrupts	27
Πίνακας Α.1 Εντολές εκτέλεσης Προγράμματος.....	89
Πίνακας Α.2 Εμφάνιση και τροποποίηση δεδομένων.....	89
Πίνακας Α.3 Διαχείριση Breakpoints.....	90
Πίνακας Α.4 Διαχείριση Προγραμμάτων	90
Πίνακας Α.5 Διαχείριση του Συστήματος.....	90

ΕΙΣΑΓΩΓΗ

Στην παρούσα διπλωματική εργασία υλοποιήθηκε προσομοιωτής ενός PWM Αντιστροφέα (PWM Inverter) με χρήση επεξεργαστή DSP. Αναπτύχθηκαν μέθοδοι Διαμόρφωσης Εύρους Παλμών, οι οποίες και πιστοποιήθηκαν σε προσομοίωση του κυκλώματος ισχύος του inverter. Οι DSP επεξεργαστές χρησιμοποιούνται κυρίως σε συστήματα επεξεργασίας σήματος και τα πλεονεκτήματά τους είναι πολλά. Σημαντικό γεγονός θεωρείται η ανάπτυξη της μεθόδου με χρήση γλώσσας προγραμματισμού assembly, κάτι που διευκόλυνε στη βαθύτερη κατανόηση της αρχιτεκτονικής του DSP. Αποτέλεσμα της εργασίας είναι η εξαγωγή ημιτονοειδούς σήματος ελεγχόμενου πλάτους και συχνότητας.

Η διπλωματική εργασία χωρίζεται σε πέντε κεφάλαια.

Το πρώτο Κεφάλαιο αναφέρεται στην οικογένεια επεξεργαστών TMS320C3x της Texas Instruments. Δίνεται λεπτομερής αναφορά στην αρχιτεκτονική τους και, κυρίως, στους πόρους που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Επίσης, αναφέρονται πλεονεκτήματα χρήσης DSPs και γενικές εφαρμογές αυτών.

Στο δεύτερο Κεφάλαιο δίνονται οι έννοιες των Ηλεκτρονικών Ισχύος και, ειδικότερα, των Inverters. Αναφέρονται με λεπτομέρεια οι PWM μέθοδοι οδήγησης των αντιστροφέων, ενώ, τέλος, περιγράφονται αναλυτικά οι μέθοδοι με Μονοπολική και Διπολική Τάση Εξόδου (Unipolar και Bipolar Voltage Switching). Το φάσμα συχνοτήτων για κάθε μία από αυτές βοηθά στην κατανόηση των διαφορών τους.

Στο τρίτο Κεφάλαιο της εργασίας γίνεται αναφορά στην υλοποίηση των παραπάνω μεθόδων. Η υλοποίηση αφορά τόσο στη μέθοδο με Μονοπολική τάση εξόδου όσο και σε αυτή με Διπολική τάση εξόδου, αλλά και στη Three-Phase PWM.

Στο τέταρτο Κεφάλαιο της εργασίας παρουσιάζεται η προσομοίωση του κυκλώματος Ισχύος, που σχεδιάστηκε ώστε να υπάρξει το επιθυμητό αποτέλεσμα, δηλαδή ημιτονοειδές σήμα εξόδου ελεγχόμενου πλάτους και συχνότητας. Παρατίθενται, επίσης, πειραματικά αποτελέσματα και μετρήσεις με διαφορετικές παραμέτρους.

Στο πέμπτο και τελευταίο Κεφάλαιο δίνονται τα συμπεράσματα της διπλωματικής εργασίας, καθώς και πιθανές μελλοντικές επεκτάσεις αυτής.

Τέλος, ακολουθεί Παράρτημα που αφορά στην αναπτυξιακή πλακέτα (Development System Kit, DSK) της Texas, που βασίζεται στον επεξεργαστή TMS320C31. Δίνεται πλήρης οδηγός χρήσης του DSK, καθώς και παραδείγματα.

ΚΕΦΑΛΑΙΟ 1^ο : ΕΙΣΑΓΩΓΗ ΣΤΗΝ TMS320C3x ΣΕΙΡΑ DSPs.

1. Εισαγωγή.

Η γενιά TMS320C3x των single-chip επεξεργαστών ψηφιακού σήματος (DSPs) της Texas Instruments, βασίζεται σε 32-bit αρχιτεκτονική. Η 'C3x οικογένεια αποτελείται από τα: 'C30, 'C31, 'LC31 και 'C32. Το ευρύτερο σύνολο στο οποίο ανήκουν και οι πιο πάνω επεξεργαστές, η TMS320 οικογένεια, έχει καθιερωθεί στο χώρο της και θεωρείται ιδανική για DSP εφαρμογές, λόγω του συνδυασμού του «δυνατού» instruction set που διαθέτει με την αρχιτεκτονική του. Με τον όρο «δυνατό» instruction set υπονοείται η ύπαρξη εξειδικευμένων εντολών, όπως αυτές που μπορούν να δηλωθούν σε ζευγάρια ώστε να εκτελεστούν παράλληλα. Λεπτομερής αναφορά στο set εντολών της C3x οικογένειας θα γίνει σε επόμενη ενότητα.

Ο maximum κύκλος των 40ns που χαρακτηρίζει τους TMS320C31-50, επιτρέπει την εκτέλεση πράξεων με ρυθμό απόδοσης που φτάνει μέχρι και τα 50 εκατομμύρια εντολές κινητής υποδιαστολής το δευτερόλεπτο (MFLOPS) και τα 25 εκατομμύρια εντολές το δευτερόλεπτο (MIPS).

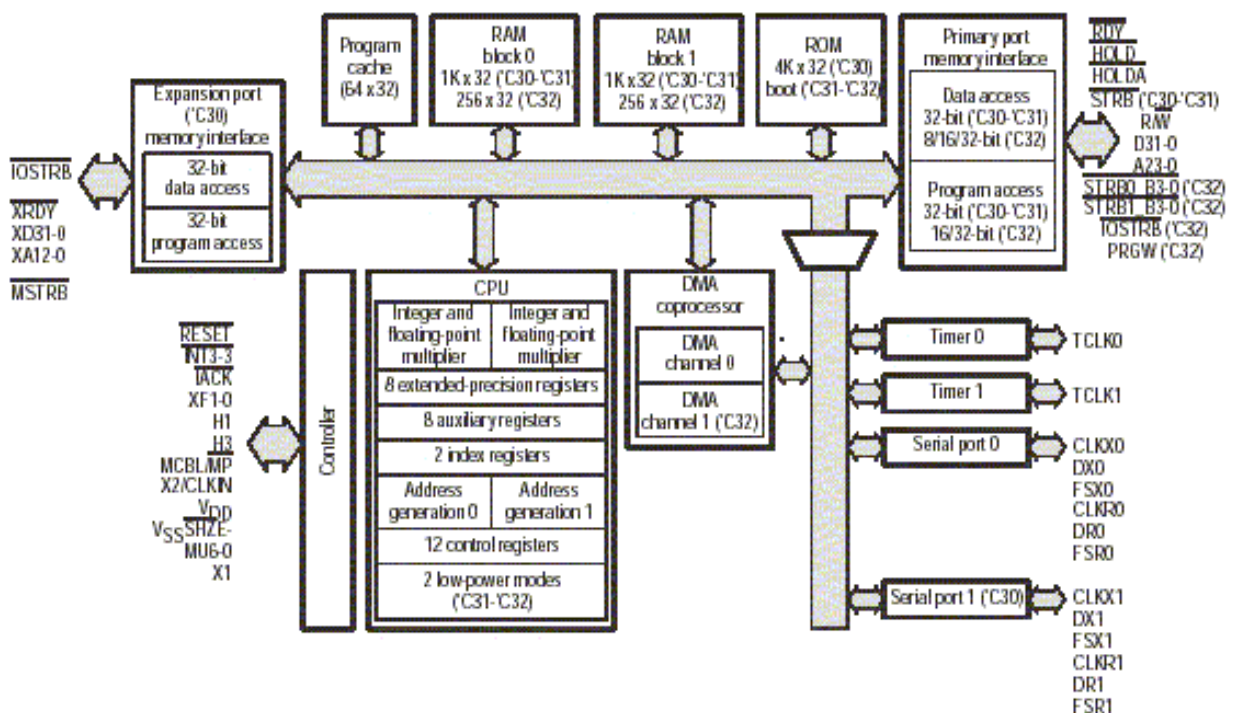
Το κύριο χαρακτηριστικό της γενιάς αυτής είναι η μεγάλης ταχύτητας επεξεργασία μαθηματικών πράξεων. Το σετ εντολών (instruction set), αλλά και το εσωτερικό datapath (busing) παρέχει μεγάλη ταχύτητα και ευελιξία. Οι TMS320C30 και TMS320C31 μπορούν να εκτελέσουν παράλληλα πράξεις πολλαπλασιασμού και ALU πράξεις με δεδομένα είτε ακέραιους ή δεδομένα κινητής υποδιαστολής σε ένα κύκλο.

Όσον αφορά στο TMS320C31, που αποτελεί και τον επεξεργαστή της παρούσας εφαρμογής, είναι ένας 32-bit αρχιτεκτονικής DSP, χαμηλού κόστους, που προσφέρει όλα τα πλεονεκτήματα των επεξεργαστών κινητής υποδιαστολής. Περιλαμβάνει:

- Δύο 32-bit timers
- Μία σειριακή θύρα, που υποστηρίζει 8-/16-/24-/32- bit μεταφορά δεδομένων
- Δύο 1K x 32bit blocks RAM, που επιτρέπει διπλή πρόσβαση σε ένα κύκλο ρολογιού
- 24-bit διευθύνσεις
- 40-/32-bit πολλαπλασιαστή και ALU κινητής υποδιαστολής/ακεραίου
- 8 καταχωρητές εκτεταμένης ακρίβειας

- Πράξεις ακεραίων, κινητής υποδιαστολής και λογικές
- Εντολές δύο και τριών τελεστών
- Παράλληλες εντολές πολλαπλασιαστή και ALU σε ένα κύκλο ρολογιού
- Διατίθεται σε 132-pin package.
- Το internal busing (επτά (7) συνολικά εσωτερικά buses για την επικοινωνία των διαφόρων υποσυστημάτων του DSP), καθώς και το ιδιαίτερο instruction set της οικογένειας αυτής παρέχει τη δυνατότητα να εκτελούνται μέχρι και 50 MFLOPS.

Το block diagram της αρχιτεκτονικής του TMS320C3x φαίνεται στο Σχ.1.1.



Σχ.1.1: Block diagram των επεξεργαστών TMS320C3x

2. Αρχιτεκτονική των 'C3x DSPs.

Η 'C3x αρχιτεκτονική μπορεί να ανταποκριθεί σε συστήματα που υλοποιούν αλγόριθμους τόσο σε hardware όσο και software. Η ακρίβεια και η δυναμική των μονάδων κινητής υποδιαστολής, η μεγάλη μνήμη που βρίσκεται στο ολοκληρωμένο, ο υψηλός βαθμός παραλληλισμού είναι μερικοί από τους λόγους στους οποίους αποδίδεται η καλή απόδοση των συγκεκριμένων DSPs.

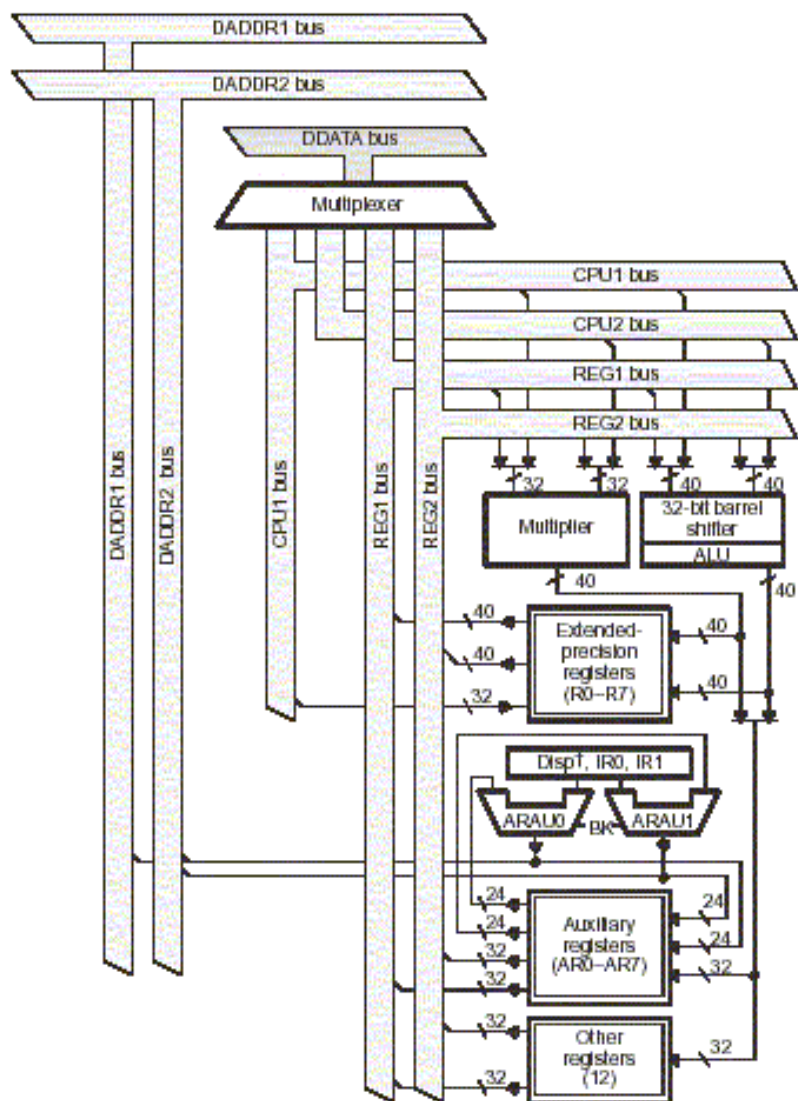
Στη συνέχεια, θα αναφερθούμε σε μερικά από τα κύρια μέρη της 'C3x αρχιτεκτονικής.

2.1 Central Processing Unit (CPU)

Η αρχιτεκτονική των 'C3x είναι βασισμένη σε καταχωρητές (register-based architecture). Αποτελείται από τα ακόλουθα στοιχεία :

- Πολλαπλασιαστή ακεραίων/δεκαδικών
- Αριθμητική και λογική μονάδα (ALU)
- 32-bit ολισθητή (shifter)
- Εσωτερικούς διαύλους (buses)
- Βοηθητικές αριθμητικές μονάδες καταχωρητών (ARAUs)

Το ακόλουθο σχήμα δείχνει ένα διάγραμμα των στοιχείων της CPU.



Σχ.1.2: Κεντρική Μονάδα Επεξεργασίας (CPU)

2.1.1 Πολλαπλασιαστής ακέραιων\δεκαδικών αριθμών

Αποδίδει πολλαπλασιασμούς σε 24-bit ακέραιους και 32-bit δεκαδικούς αριθμούς σε ένα κύκλο ρολογιού και επιτρέπει πράξεις σταθερής ή κινητής υποδιαστολής έως και 33ns ανά κύκλο εντολής. Το throughput αυξάνεται στην περίπτωση παράλληλων εντολών, δηλαδή εκτέλεση πολλαπλασιασμού και άλλης ALU πράξης σε ένα κύκλο. Όταν εκτελείται πολλαπλασιασμός κινητής υποδιαστολής, τα δεδομένα εισόδου είναι 32-bit δεκαδικοί αριθμοί και η έξοδος είναι 40-bit αριθμός. Στην περίπτωση πολλαπλασιασμού ακεραίων, η είσοδος είναι 24-bit και η έξοδος 32-bit.

2.1.2 Αριθμητική και λογική μονάδα και εσωτερικά buses(δίαυλοι).

Η ALU αποδίδει πράξεις σε 32-bit ακέραια, 32-bit λογικά και 40-bit κινητής υποδιαστολής (floating-point) δεδομένα σε ένα κύκλο, ενώ τα αποτελέσματα είναι 32-bit ακέραιοι ή 40-bit δεκαδικοί αριθμοί.

Τα τέσσερα εσωτερικά buses, τα CPU1, CPU2, REG1 και REG2, μεταφέρουν δύο τελεστές από τη μνήμη και δύο από το αρχείο καταχωρητών, επιτρέποντας, έτσι, παράλληλους πολλαπλασιασμούς και προσθέσεις\αφαιρέσεις σε τέσσερις ακέραιους ή δεκαδικούς τελεστές σε ένα κύκλο ρολογιού.

2.1.3 Βοηθητικές αριθμητικές μονάδες καταχωρητών (ARAUs).

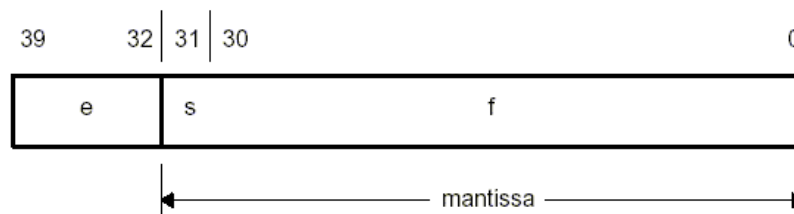
Δύο βοηθητικές αριθμητικές μονάδες καταχωρητών (auxiliary register arithmetic units), οι ARAU0 και ARAU1, έχουν τη δυνατότητα να παράγουν δύο διευθύνσεις σε ένα κύκλο. Οι ARAUs λειτουργούν παράλληλα με τον πολλαπλασιαστή και την ALU.

2.2 Πρωτεύον αρχείο καταχωρητών της CPU (CPU Primary Register File).

Η 'C3x αρχιτεκτονική παρέχει 28 καταχωρητές σε ένα block, το οποίο είναι άμεσα συνδεδεμένο με τη CPU. Όλοι αυτοί οι καταχωρητές μπορούν να χρησιμοποιηθούν από τον πολλαπλασιαστή και την ALU ως καταχωρητές γενικής χρήσης (general-purpose registers). Μπορούν, όμως, να έχουν και ειδική χρήση, όπως για παράδειγμα, οι οκτώ εκτεταμένης ακρίβειας (extended-precision) καταχωρητές, που είναι κατάλληλοι για τη διατήρηση εκτεταμένης ακρίβειας (extended-precision) αποτελεσμάτων κινητής υποδιαστολής. Οι οκτώ αυτοί καταχωρητές υποστηρίζουν έμμεση διευθυνσιοποίηση και μπορούν να χρησιμοποιηθούν και ως 32-bit (ακέραιοι και λογικοί) καταχωρητές γενικής χρήσης. Οι υπόλοιποι παρέχουν στο σύστημα λειτουργίες όπως διευθυνσιοποίηση, διαχείριση στοίβας, πληροφορίες για την κατάσταση του επεξεργαστή, interrupts, κ.ά.

Πιο ειδικά :

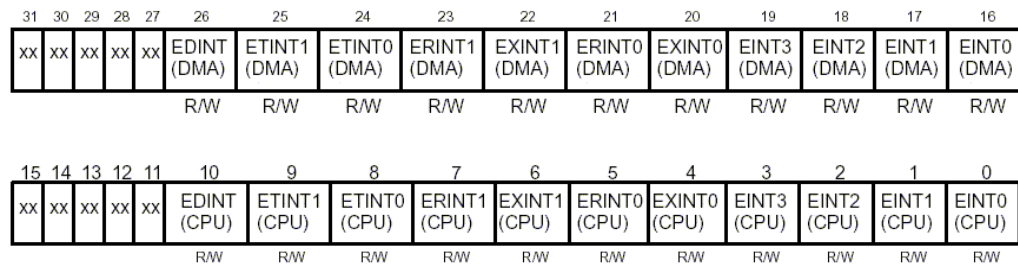
- Οι **extended-precision καταχωρητές (R7-R0)** μπορούν να αποθηκεύσουν και να υποστηρίξουν πράξεις σε 32-bit ακέραιους και 40-bit floating point αριθμούς. Κάθε εντολή της οποίας οι τελεστές είναι αριθμοί κινητής υποδιαστολής, χρησιμοποιεί τα bits 39-0. Εάν οι τελεστές είναι ακέραιοι (με ή χωρίς πρόσημο), χρησιμοποιούνται τα bits 31-0, ενώ τα bits 39-32 παραμένουν αμετάβλητα. Πιο ειδικά, τα 39-32 bits χρησιμοποιούνται για την αποθήκευση του εκθετικού (e) ενός αριθμού κινητής υποδιαστολής, ενώ τα 31-0 bits αποθηκεύουν το κυρίως μέρος (mantissa) του αριθμού, όπως φαίνεται και στο Σχ.1.3.



Σχ.1.3: Διάταξη καταχωρητή εκτεταμένης ακρίβειας

- Οι **32-bit βοηθητικοί καταχωρητές (AR7-AR0)** είναι προσβάσιμοι από τη CPU και χρησιμοποιούνται από τις δύο ARAUs. Η κύρια λειτουργία των καταχωρητών αυτών είναι η παραγωγή 24-bit διευθύνσεων. Μπορούν, επίσης, να χρησιμοποιηθούν ως μετρητές σε επαναλήψεις βρόχων, ή ακόμη και σαν 32-bit καταχωρητές γενικής χρήσης από τον πολλαπλασιαστή και την ALU.
- Ο **δείκτης σελίδας δεδομένων (data page pointer, DP)** είναι ένας 32-bit καταχωρητής. Τα οκτώ least significant bits του καταχωρητή χρησιμοποιούνται για άμεση διευθυνσιοποίηση, δηλαδή ο καταχωρητής χρησιμοποιείται ως pointer στην σελίδα των δεδομένων των οποίων έχει δοθεί η διεύθυνση. Οι σελίδες δεδομένων έχουν μέγεθος 64K και είναι 256 συνολικά.
- Οι 32-bit **index registers (IR0,IR1)** περιέχουν την τιμή που χρησιμοποιεί η ARAU για τον υπολογισμό indexed διεύθυνσης.
- Επιπλέον, η ARAU χρησιμοποιεί τον **καταχωρητή μεγέθους μπλοκ (block size register, BK)** στην κυκλική διευθυνσιοποίηση για τον καθορισμό του μεγέθους του block.

- Ο 32-bit δείκτης στοίβας συστήματος (**system-stack pointer, SP**) περιέχει την διεύθυνση της κορυφής της στοίβας του συστήματος. Πάντοτε δείχνει στο τελευταίο στοιχείο που έχει μπει στη στοίβα. Ένα push προκαλεί αύξηση (κατά 1) του καταχωρητή, αφού πρώτα γίνει η αποθήκευση, ενώ, ένα pop προκαλεί μείωση (κατά 1) του pointer, προτού γίνει η ανάκτηση. Είναι κατανοητό πως ο SP έχει άμεση σχέση με τα interrupts.
- Ο καταχωρητής κατάστασης (**status register, ST**) περιέχει πληροφορίες για την κατάσταση της CPU. Οι πράξεις, όπως αριθμητικές ή λογικές συναρτήσεις, θέτουν τα condition flags του ST, ανάλογα με το αν το αποτέλεσμα τους είναι 0, αρνητικό, κ.ά. Μετά από μία πράξη load, το περιεχόμενο του καταχωρητή κατάστασης είναι ίδιο με αυτό του source operand της πράξης. Αυτό γίνεται με bit προς bit αντικατάσταση του περιεχομένου του ST.
- Ο **interrupt-enable register (IE)** είναι ένας 32-bit καταχωρητής, που όταν περιέχει την τιμή 1 σε κάποιο από τα bits του, ενεργοποιείται το αντίστοιχο interrupt. Τα bits στις θέσεις 10-0 αφορούν στην ενεργοποίηση των CPU interrupts, ενώ τα bits στις θέσεις 26-16 αφορούν στα DMA interrupts, όπως φαίνεται και στο Σχήμα 1.4. Στο reset, στον IE register γράφεται μηδέν (0).



xx = reserved bit, διαβάζεται ως 0.

Σχ.1.4: Interrupt Enable Register (IE)

- Ο **interrupt flag register (IF)** είναι ένας, επίσης, 32-bit καταχωρητής. Για να θέσουμε ένα interrupt, κάνουμε 1 το interrupt flag register bit. Στην περίπτωση που αυτό είναι μηδέν (0), τότε το αντίστοιχο interrupt δεν έχει τεθεί.
- Ο **IO flag register (IOF)** ελέγχει τη λειτουργία των εξωτερικών pins του DSP, των XF0 και XF1. Τα pins αυτά μπορούν να είναι είσοδος ή έξοδος (input ή output) και να γίνονται σε αυτά read ή write. Στο Σχ.1.5 καθώς και

τον Πίνακα 1.1 φαίνονται η δομή του IOF register, όπως και συνοπτική περιγραφή των bits αυτού.



Σημ. 1) xx = reserved bit, διαβάζεται ως 0
2) R = read, W = write

Σχ.1.5: I/O-Flag Register (IOF)

Bit	Name	Reset Value	Function
0	Reserved	0	Read as 0
1	I/OXF0	0	If I/OXF0 = 0, XF0 is configured as a general-purpose input pin. If I/OXF0 = 1, XF0 is configured as a general-purpose output pin.
2	OUTXF0	0	Data output on XF0
3	INXF0	0	Data input on XF0. A write has no effect.
4	Reserved	0	Read as 0
5	I/OXF1	0	If I/OXF1 = 0, XF1 is configured as a general-purpose input pin. If I/OXF1 = 1, XF1 is configured as a general-purpose output pin.
6	OUTXF1	0	Data output on XF1
7	INXF1	0	Data input on XF1. A write has no effect.
31–8	Reserved	0–0	Read as 0

Πίνακας 1.1: Περίληψη των bits του IOF Register

Στην παρούσα εφαρμογή, η σύγκριση των δύο σημάτων θα μπορούσε εναλλακτικά να αναπαρασταθεί με τη βοήθεια του IOF register. Δηλαδή, όταν το αποτέλεσμα είναι ίσο με 0, τότε να φορτώνεται στον IOF, και συγκεκριμένα στο XF0 pin, η τιμή 0. Στην αντίθετη περίπτωση, το XF0 να έχει την τιμή 1. Με αυτό τον τρόπο, το αποτέλεσμα θα εμφανίζεται ως έξοδος του DSP, στα XF0 και XF1 pins του επεξεργαστή.

- Ο **μετρητής επανάληψης (repeat counter, RC)** είναι ένας 32-bit καταχωρητής, ο οποίος καθορίζει τον αριθμό των φορών που θα επαναληφθεί ένα block. Όταν ο επεξεργαστής λειτουργεί σε κατάσταση επανάληψης, ο 32-bit repeat start-address register (RS) περιέχει την αρχική διεύθυνση του μέρους

του προγράμματος (block) που πρόκειται να επαναληφθεί, ενώ, ο 32-bit repeat end-address register (RE) περιέχει την τελική διεύθυνση του block αυτού.

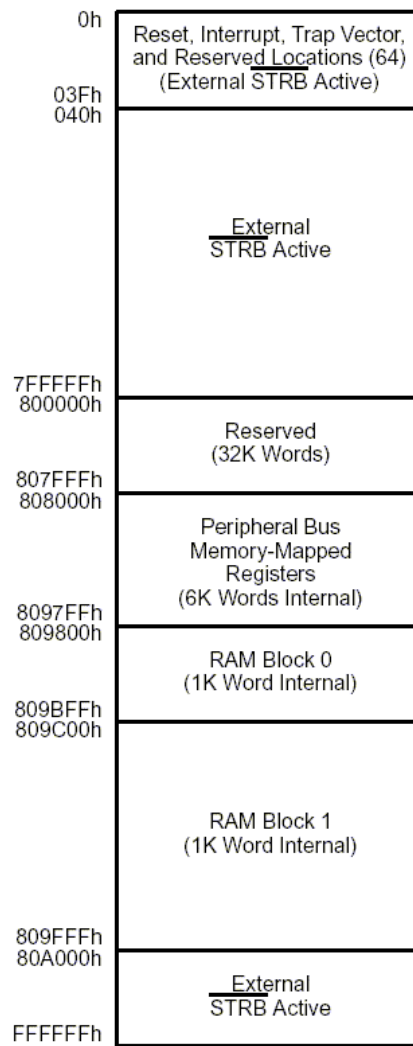
2.3 Άλλοι καταχωρητές.

Στην παραπάνω ενότητα είδαμε, εν συντομία, τους καταχωρητές του Primary Register File. Παρακάτω, θα δούμε δύο επιπλέον καταχωρητές, εξίσου σημαντικούς για την κατανόηση της 'C3x αρχιτεκτονικής.

- Ο **program counter (PC)** είναι ένας 32-bit καταχωρητής, που περιέχει τη διεύθυνση της επόμενης εντολής που πρόκειται να γίνει fetch.
- Ο **instruction register (IR)** είναι κι αυτός ένας 32-bit καταχωρητής που κρατά το opcode της εντολής, κατά τη διάρκεια της αποκωδικοποίησης αυτής. Ο IR δεν είναι προσβάσιμος στη CPU.

2.4 Οργάνωση μνήμης.

Ο συνολικός χώρος μνήμης του 'C3x είναι 16Mx32-bit words, δηλαδή έχει δυνατότητα διευθυνσιοδότησης 64Mbytes. Το πρόγραμμα, τα δεδομένα, αλλά και το I/O περιέχονται σε αυτόν τον χώρο μνήμης. Το περιεχόμενο των πινάκων, των συντελεστών, ο κώδικας του προγράμματος, και γενικά, τα δεδομένα, μπορούν να αποθηκευτούν είτε στη RAM ή στη ROM. Στην περίπτωση του 'C31, on-chip υπάρχει μόνο μνήμη RAM. Ο χώρος μνήμης των 'C3x επεξεργαστών προκύπτει από το γεγονός ότι έχουν 24-bit χώρο διευθύνσεων και 32-bit data. Φυσικά, στο χώρο μνήμης περιέχονται και τα απαραίτητα για τον καθορισμό των interrupts, των registers, κ.ο.κ.

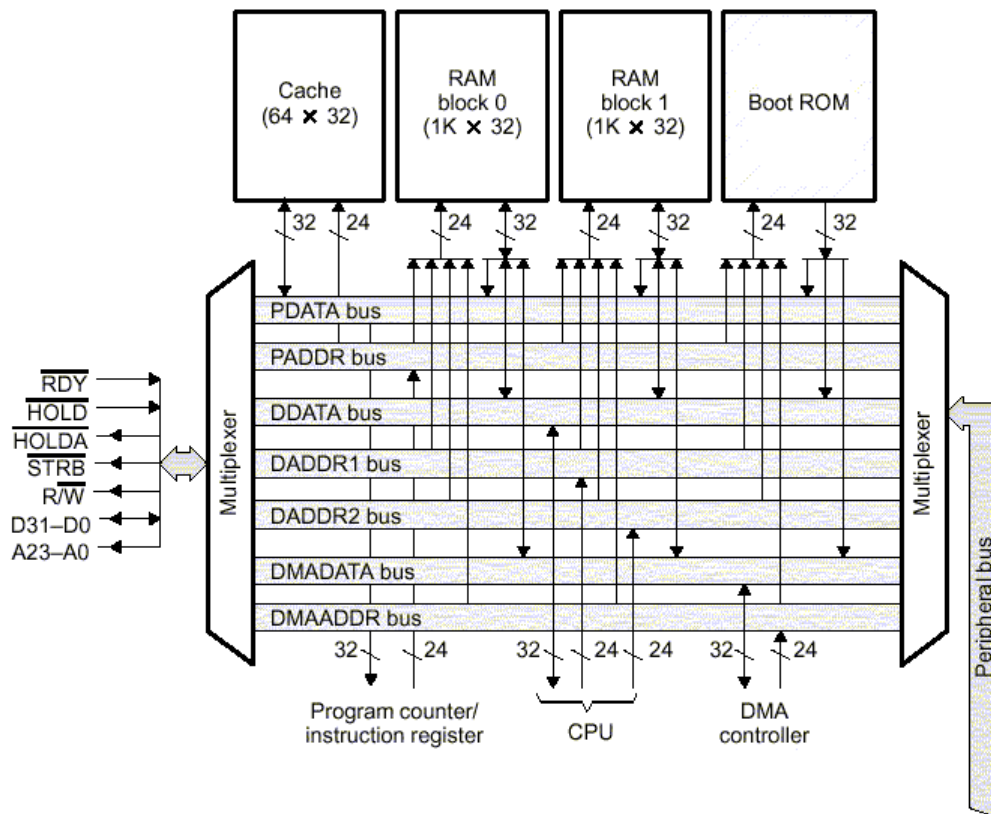


Σχ.1.6: Memory map του TMS320C31

Η RAM χωρίζεται σε δύο blocks, τα block0 και block1, 1K x 32 bits το καθένα. Έτσι, επιτρέπει δύο προσβάσεις σε ένα κύκλο ρολογιού. Το παραπάνω σχήμα, Σχ.1.6, δείχνει την οργάνωση της μνήμης στον TMS320C31.

2.5 Λειτουργία εσωτερικού bus.

Η υψηλή απόδοση της 'C3x οικογένειας επεξεργαστών οφείλεται σε μεγάλο βαθμό στο εσωτερικό datapath (busing) αυτών και τη δυνατότητα παραλληλισμού. Ξεχωριστοί δίαυλοι (buses) επιτρέπουν παράλληλα fetches προγράμματος, προσβάσεις σε δεδομένα και DMA προσβάσεις. Τα buses αυτά συνδέουν τις on-chip και off-chip μνήμες καθώς και τα περιφερειακά του συστήματος. Στο Σχ.1.7 φαίνονται αναλυτικά μαζί με τις συνδέσεις τους με τα memory blocks.



Σχ.1.7: Εσωτερικό busing του TMS320C31

Έτσι, τα 24-bit data address busses (DADDR1 και DADDR2) και το 32-bit data data bus (DDATA) υποστηρίζουν δύο data-memory προσβάσεις σε κάθε κύκλο. Το DDATA bus μεταφέρει δεδομένα στη CPU μέσω των CPU1 και CPU2 buses. Αυτά μπορούν να μεταφέρουν δύο memory-data τελεστές στον πολλαπλασιαστή, την ALU και το αρχείο καταχωρητών σε κάθε κύκλο.

Επίσης, εσωτερικά στη CPU υπάρχουν τα Register buses REG1 και REG2, όπως φαίνεται στο Σχ.1.2, τα οποία μπορούν να μεταφέρουν δύο τιμές δεδομένων από τον πολλαπλασιαστή και την ALU σε κάθε κύκλο.

Τέλος, ο DMA controller υποστηρίζεται από το 24-bit address bus (DMAADDR) και από το 32-bit data bus (DMADATA). Αυτά τα buses επιτρέπουν στη DMA να έχει προσβάσεις στη μνήμη παράλληλα με προσβάσεις σε αυτή μέσω των data και program buses.

2.6 Λειτουργία εξωτερικού bus.

Το TMS320C31 παρέχει ένα εξωτερικό interface: το primary bus, ώστε να έχει πρόσβαση σε εξωτερική μνήμη και περιφερειακές συσκευές. Αποτελείται από ένα 32-bit bus

δεδομένων (data bus) και ένα 24-bit bus διευθύνσεων (address bus). Μπορεί να χρησιμοποιηθεί για I/O προσβάσεις, όπως και έγινε στην παρούσα εφαρμογή.

Με το primary bus σχετίζεται ο primary-bus control register, του οποίου η διεύθυνση είναι η 808064h. Είναι ένας 32-bit register, ο οποίος περιέχει τα control bits του primary bus.

Στην ανάλυση του κώδικα, θα γίνει περιγραφή του τρόπου με τον οποίο χρησιμοποιήθηκε το primary bus ώστε οι παλμοί εξόδου της SPWM μεθόδου να «βγαίνουν» ως εξοδοι του DSP.

2.7 Περιφερειακά.

Τα περιφερειακά του TMS320C31 DSP περιλαμβάνουν δύο timers και μία σειριακή θύρα. Ελέγχονται μέσω memory-mapped καταχωρητών στο peripheral bus του DSP. Αυτό αποτελείται από ένα 32-bit bus για δεδομένα και ένα 24-bit bus για διευθύνσεις.

2.7.1 Timers.

Το TMS320C31 έχει δύο timers, δύο 32-bit μετρητές με εσωτερικό ή εξωτερικό χρονισμό. Μπορούν να «χτυπήσουν» εσωτερικά στο 'C31, ή εξωτερικά του συστήματος σε δεδομένα χρονικά διαστήματα, ή και να λειτουργήσουν ως απλοί μετρητές (counters). Κάθε timer έχει ένα I/O pin, το οποίο μπορεί να χρησιμοποιηθεί σαν εσωτερικό ρολόι του timer, σαν εξωτερικό σήμα, οδηγούμενο από τον timer, ή ως ένα I/O pin γενικής χρήσης.

Για κάθε timer, χρησιμοποιούνται τρεις memory-mapped registers:

- Global-Control Register

Καθορίζει την κατάσταση του timer και ελέγχει το I/O pin του timer. Βρίσκεται στη θέση μνήμης 808020 για τον Timer0 και στην θέση 808030 για τον Timer1, όπως φαίνεται και στο Σχ.5.

- Period Register

Καθορίζει τη συχνότητα του timer. Βρίσκεται στη θέση μνήμης 808028 για τον Timer0 και στην θέση 808038 για τον Timer1, όπως φαίνεται και στο Σχ.1.5.

- Counter Register

Περιέχει τη τρέχουσα τιμή του counter. Ο counter μηδενίζεται και προκαλεί interrupt μόλις η τιμή του γίνει ίση με αυτή του Period Register. Βρίσκεται στη θέση μνήμης 808024 για τον Timer0 και στην θέση 808034 για τον Timer1, όπως φαίνεται και στο Σχ.1.8.

808000h	DMA Global Control
808004h	DMA Source Address
808006h	DMA Destination Address
808008h	DMA Transfer Counter
808020h	Timer 0 Global Control
808024h	Timer 0 Counter
808028h	Timer 0 Period Register
808030h	Timer 1 Global Control
808034h	Timer 1 Counter
808038h	Timer 1 Period Register
808040h	Serial Global Control
808042h	FSX/DX/CLKX Serial Port Control
808043h	FSR/DR/CLKR Serial Port Control
808044h	Serial R/X Timer Control
808045h	Serial R/X Timer Counter
808046h	Serial R/X Timer Period Register
808048h	Data-Transmit
80804Ch	Data-Receive
808064h	Primary-Bus Control

Σχ.1.8: Peripheral bus memory-mapped registers

Όπως προαναφέρθηκε, οι timers του επεξεργαστή μπορούν να χρησιμοποιηθούν για να παράγουν interrupts, όπως έγινε και στην παρούσα εφαρμογή. Όταν «χτυπήσει» ένα interrupt, σταματά η κανονική διαδικασία διεξαγωγής της τρέχουσας εφαρμογής και αρχίζει να εκτελείται η ISR – Interrupt Service Routine. Η ρουτίνα αυτή, αρχικά, αποθηκεύει τις τιμές των registers, και γενικά όλα τα σημαντικά δεδομένα μέχρι τη στιγμή της διακοπής, σε μία stack. Κατόπιν, εκτελείται η ρουτίνα του interrupt και, μόλις αυτή τελειώσει, γίνεται η επαναφορά των δεδομένων από τη stack, ώστε να συνεχιστεί η εκτέλεση από το σημείο που έγινε η διακοπή.

Το ακόλουθο σχήμα (Σχ.1.9) δείχνει τις θέσεις μνήμης των interrupt service routines, οι οποίες αρχίζουν να εκτελούνται όταν συμβεί interrupt. Η ρουτίνα TINT0 αφορά στον Timer0, ενώ η TINT1 αφορά στον Timer1.

809FC1h	INT0
809FC2h	INT1
809FC3h	INT2
809FC4h	INT3
809FC5h	XINT0
809FC6h	RINT0
809FC7h	Reserved
809FC8h	
809FC9h	TINT0
809FCAh	TINT1
809FCBh	DINT
809FCCh	Reserved
809FDFh	
809FE0h	TRAP 0
	•
	•
	•
809FFBh	TRAP 27
809FFCh	Reserved
809FFh	

Σχ.1.9: Θέσεις μνήμης των interrupts

Ένα timer interrupt παράγεται όποτε το TSTAT bit του control register αλλάξει από 0 σε 1. Η συχνότητά του εξαρτάται από το εάν ο timer έχει τεθεί σε pulse ή clock mode.

- Σε pulse mode, η συχνότητα του interrupt καθορίζεται από την ακόλουθη εξίσωση:

$$f_{(\text{interrupt})} = \frac{f_{(\text{timer_clock})}}{\text{period_register}}, \text{ όπου}$$

$$f_{(\text{interrupt})} = \text{timer_frequency}$$

$$f_{(\text{timer_clock})} = \text{interrupt_frequency}$$

- Σε clock mode, η συχνότητα του interrupt καθορίζεται από την ακόλουθη εξίσωση:

$$f_{(\text{interrupt})} = \frac{f_{(\text{timer_clock})}}{2 \times \text{period_register}}, \text{ όπου}$$

$$f_{(\text{interrupt})} = \text{timer_frequency}$$

$$f_{(\text{timer_clock})} = \text{interrupt_frequency}$$

Ο μετρητής του timer κάνει αυτόματα reset στην τιμή 0, όποτε γίνει ίσος με την τιμή του timer period register. Το σήμα ελέγχου για την ενεργοποίηση interrupt για κάθε από τους δύο timers, είτε πρόκειται για CPU ή για DMA, βρίσκεται στον CPU/DMA interrupt enable register.

Οι δύο timers του TMS320C31 χρησιμοποιήθηκαν στον κώδικα του Single-Phase SPWM ως interrupts. Στις ρουτίνες αυτών παράγονται το ημίτονο και ο τριγωνικός παλμός, που είναι απαραίτητα για την υλοποίηση της μεθόδου. Η σωστή ενεργοποίηση των timers, με βάση τα παραπάνω είναι πρωταρχικής σημασίας. Περισσότερα για τον τρόπο που αυτοί λειτουργούν, θα αναφερθούν στην ανάλυση του κώδικα.

2.7.2 Σειριακή Θύρα.

Το 'C31 έχει μία σειριακή θύρα διπλής κατεύθυνσης. Έχει τη δυνατότητα να μεταφέρει 8, 16, 24, ή 32 bits δεδομένων. Το ρολόι της σειριακής θύρας μπορεί να είναι είτε εσωτερικό ή εξωτερικό.

3. Διευθυνσιοδότηση (Addressing).

Η TMS320C3x οικογένεια υποστηρίζει πέντε τρόπους διευθυνσιοδότησης, μεταξύ των οποίων είναι η μέθοδος γενικής διευθυνσιοδότησης, η μέθοδος διευθυνσιοδότησης τριών τελεστών και η παράλληλη διευθυνσιοδότηση. Έξι τύποι αυτής μπορούν να χρησιμοποιηθούν σε κάθε μία από τις παραπάνω μεθόδους, οι οποίοι επιτρέπουν πρόσβαση σε δεδομένα μνήμης και καταχωρητών. Οι έξι τύποι διευθυνσιοδότησης είναι οι:

- Καταχωρητή (Register)
- Άμεση (Direct)
- Έμμεση (Indirect)
- Short-Immediate
- Long-Immediate
- PC-Relative

Βέβαια, κάποιοι από τους παραπάνω τύπους είναι κατάλληλοι για κάποιες εντολές και κάποιοι όχι.

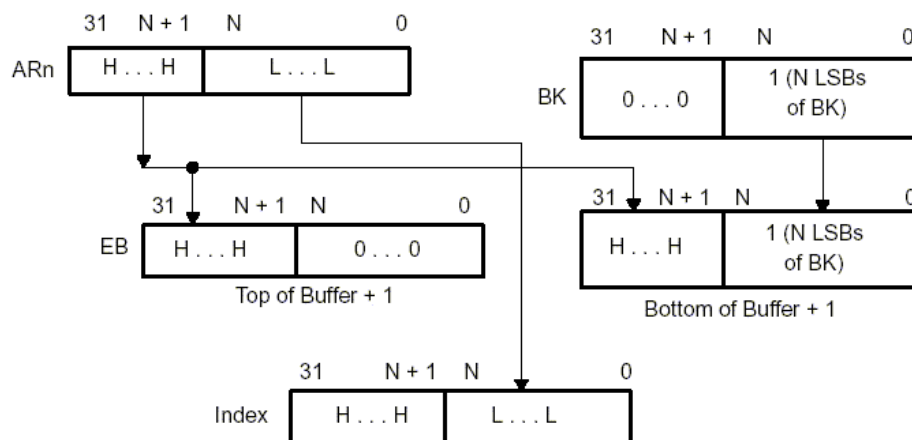
3.1 Κυκλική διευθυνσιοδότηση (Circular Buffering).

Στην ενότητα αυτή, θα γίνει ανάλυση μίας ιδιαίτερης μεθόδου διευθυνσιοδότησης, της κυκλικής, μιας και έγινε ευρεία χρήση αυτής κατά την ανάπτυξη του κώδικα της παρούσας εργασίας.

Πολλοί αλγόριθμοι, όπως η συνέλιξη (convolution), απαιτούν την υλοποίηση ενός κυκλικού buffer στη μνήμη, ο οποίος περιέχει τα πιο πρόσφατα στοιχεία που υποβάλλονται σε επεξεργασία. Καθώς έρχονται καινούρια δεδομένα, αυτά αντικαθιστούν τα παλαιότερα.

Ο block size καταχωρητής (BK) καθορίζει το μέγεθος του κυκλικού buffer. Ονομάζοντας το most significant 1 του καταχωρητή BK ως bit N, με $N \leq 15$, μπορούμε να βρούμε τη διεύθυνση που θέλουμε, ακολουθώντας τη βάση του κυκλικού buffer, με τον εξής τρόπο: συνενώνουμε τα bits 31 έως N+1 ενός από τους βοηθητικούς καταχωρητές ARn με τα bits N έως 0 του BK καταχωρητή. Η διεύθυνση της κορυφής του buffer αναφέρεται ως effective base (EB) και ισοδυναμεί με τη συνένωση των 31 έως (N+1) bits του ARn με τα N έως 0 bits του EB που είναι μηδέν (0).

Ένας κυκλικός buffer, μεγέθους R, πρέπει να ξεκινά από το K bit, δηλαδή, τα K LSBs της αρχικής διεύθυνσης του κυκλικού buffer πρέπει να είναι μηδενικά. Ο K είναι ακέραιος και ισχύει γι' αυτόν: $2^K > R$. Μιας και η τιμή R πρέπει να φορτωθεί στον καταχωρητή BK, $K \geq N + 1$. Για παράδειγμα, ένας 31-word κυκλικός buffer πρέπει να ξεκινά από μία διεύθυνση, της οποίας τα 5 LSBs να είναι μηδενικά, δηλαδή XXXXXXXXXXXXXXXXXXXXXXXXXXXX00000, ενώ η τιμή 31 πρέπει να φορτωθεί στον BK. Το Σχήμα 1.10 παρουσιάζει πιο αναλυτικά τον τρόπο με τον οποίο καθορίζονται τα παραπάνω.



Σχ.1.10: Παραγωγή των index και top, bottom του κυκλικού buffer

Στην κυκλική διευθυνσιοδότηση, η λέξη `index` αναφέρεται στα `N` LSBs του βοηθητικού καταχωρητή που έχει επιλεγεί, ενώ η λέξη `step`, αφορά στην ποσότητα που πρέπει να προστεθεί ή να αφαιρεθεί από τον `ARn`. Χρησιμοποιούνται δύο κανόνες:

- Το `step` που χρησιμοποιείται πρέπει να είναι μικρότερο ή ίσο με το μέγεθος του `block`.
- Την πρώτη φορά που διευθυνσιοδοτείται η κυκλική ουρά, ο βοηθητικός καταχωρητής πρέπει να δείχνει σε ένα στοιχείο της κυκλικής ουράς.

Ο αλγόριθμος είναι ο εξής:

If $0 \leq \text{index} + \text{step} < \text{BK}$:

$\text{index} = \text{index} + \text{step}$.

Else if $\text{index} + \text{step} \geq \text{BK}$:

$\text{index} = \text{index} + \text{step} - \text{BK}$.

Else if $\text{index} + \text{step} < 0$:

$\text{index} = \text{index} + \text{step} + \text{BK}$.

Ο κυκλικός buffer είναι ιδιαίτερα χρήσιμος σε περιπτώσεις όπου απαιτείται περιοδική επανάληψη ενός συνόλου τιμών, όπως, για παράδειγμα, στην αναπαράσταση ενός περιοδικού σήματος.

4. Interrupts.

Η TMS320C3x οικογένεια και, κατά συνέπεια, και ο TMS320C31 επεξεργαστής, υποστηρίζουν πολλαπλά εσωτερικά και εξωτερικά interrupts, τα οποία μπορούν να χρησιμοποιηθούν σε μία πληθώρα εφαρμογών. Στην παρούσα ενότητα, θα συζητηθεί η λειτουργία αυτών των interrupts.

Στον Πίνακα 1.2, παρουσιάζονται οι διευθύνσεις των interrupt service routines, οι οποίες αρχίζουν να εκτελούνται όταν προκύψει κάποιο interrupt.

Address	Description
809FC1	$\overline{\text{INT0}}$
809FC2	$\overline{\text{INT1}}$
809FC3	$\overline{\text{INT2}}$
809FC4	$\overline{\text{INT3}}$
809FC5	$\overline{\text{XINT0}}$
809FC6	$\overline{\text{RINT0}}$
809FC7	Reserved
809FC8	Reserved
809FC9	$\overline{\text{TINT0}}$
809FCA	$\overline{\text{TINT1}}$
809FCB	$\overline{\text{DINT0}}$
809FCC–809FDF	Reserved

Πίνακας 1.2: Διευθύνσεις των Interrupt service routines του 'C31

Όσον αφορά στην προτεραιότητα των interrupts, όταν προκύψουν δύο στον ίδιο κύκλο ρολογιού, ή αν δύο interrupts, που έχουν ήδη φτάσει, περιμένουν να εξυπηρετηθούν, τότε κάποιο από αυτά θα εξυπηρετηθεί πριν από το άλλο. Η CPU χειρίζεται την κατάσταση αυτή, εξυπηρετώντας το interrupt με τη μικρότερο βαθμό προτεραιότητας. Στον Πίνακα 1.3, παρουσιάζονται οι βαθμοί προτεραιότητας που έχουν ανατεθεί στο reset και στα interrupts.

Reset ή Interrupt	Θέση	Βαθμός Προτ/τας	Λειτουργία
$\overline{\text{RESET}}$	0h	0	External reset signal input on the RESET pin
$\overline{\text{INT0}}$	1h	1	External interrupt on the INT0 pin
$\overline{\text{INT1}}$	2h	2	External interrupt on the INT1 pin
$\overline{\text{INT2}}$	3h	3	External interrupt on the INT2 pin
$\overline{\text{INT3}}$	4h	4	External interrupt on the INT3 pin
XINT0	5h	5	Internal interrupt generated when serial-port 0 transmit buffer is empty
RINT0	6h	6	Internal interrupt generated when serial-port 0 receive buffer is full
XINT1†	7h	7	Internal interrupt generated when serial-port 1 transmit buffer is empty
RINT1†	8h	8	Internal interrupt generated when serial-port 1 receive buffer is full
TINT0	9h	9	Internal interrupt generated by timer 0
TINT1	0Ah	10	Internal interrupt generated by timer 1
DINT	0Bh	11	Internal interrupt generated by DMA controller 0

Πίνακας 1.2: Προτεραιότητες των Interrupts

Τέσσερις καταχωρητές της CPU περιέχουν τα bits ελέγχου της λειτουργίας των interrupts:

- Status Register (ST)

Το CPU global interrupt enable bit (GIE), που βρίσκεται στον ST της CPU, ελέγχει όλα τα interrupts αυτής. Όταν το bit αυτό έχει την τιμή 1, η CPU ανταποκρίνεται σε ένα ενεργό (enabled) interrupt, ενώ, όταν το παραπάνω bit είναι μηδέν, όλα τα CPU interrupts είναι ανενεργά (disabled).
- CPU/DMA Interrupt Enable Register (IE)

Ο καταχωρητής αυτός ενεργοποιεί/απενεργοποιεί τα CPU και DMA (εξωτερικά, σειριακής θύρας και timer) interrupts.
- CPU Interrupt Flag Register (IF)

Περιέχει interrupt flag bits, που δηλώνουν ότι το αντίστοιχο interrupt έχει τεθεί.
- DMA Global Control Register

Τα DMA interrupts ελέγχονται από τα bits συγχρονισμού του καταχωρητή που περιγράφουμε. Τα interrupts αυτά είναι ανεξάρτητα από το ST (GIE) bit.

Μπορούμε να διαβάσουμε ή να γράψουμε στα bits του IF καταχωρητή με τη βοήθεια software. Γράφοντας 1 σε κάποιο από τα IF bits, θέτουμε στο flag του αντίστοιχου interrupt

την τιμή 1, ενώ, γράφοντας 0, του κάνουμε reset. Έτσι, μπορούμε να επηρεάσουμε την κατάσταση όλων των interrupts με τη χρήση software.

5. Εντολές της γλώσσας Assembly.

Το σετ εντολών της γλώσσας assembly για την TMS320C3x γενιά επεξεργαστών μπορεί να υποστηρίξει γενικής χρήσης εφαρμογές επεξεργασίας σήματος. Οι πιο πολλές από αυτές απαιτούν ένα κύκλο ρολογιού για την εκτέλεσή τους. Το instruction set περιλαμβάνει 113 εντολές, οι οποίες μπορούν να χωριστούν στις ακόλουθες ομάδες:

- Load και store
- Αριθμητικές / Λογικές δύο τελεστών
- Αριθμητικές / Λογικές τριών τελεστών
- Ελέγχου του προγράμματος
- Παράλληλες εντολές

5.1 Load και Store εντολές

Η C3x οικογένεια υποστηρίζει 13 load και store εντολές. Αυτές μπορούν να:

- φορτώσουν ένα word από τη μνήμη σε ένα καταχωρητή
- αποθηκεύσουν ένα word από ένα καταχωρητή στη μνήμη
- διαχειριστούν δεδομένα στη στοίβα του συστήματος.

Δύο από αυτές, η *LDIcond* και η *LDFcond*, μπορούν να φορτώσουν δεδομένα υπό συνθήκη. Αυτό είναι χρήσιμο για εύρεση μικρότερης ή μεγαλύτερης τιμής σε ένα σετ δεδομένων.

5.2 Εντολές δύο τελεστών.

Υποστηρίζονται 35 αριθμητικές και λογικές εντολές δύο τελεστών. Οι δύο αυτοί τελεστές είναι η προέλευση και ο προορισμός (source και destination). Ο τελεστής προέλευσης μπορεί να είναι ένα word μνήμης ή ένας καταχωρητής, ενώ ο τελεστής προορισμού είναι πάντοτε καταχωρητής. Οι εντολές αυτές παρέχουν λογικές και κινητής υποδιαστολής πράξεις καθώς και πράξεις για ακέραιους.

5.3 Εντολές τριών τελεστών.

Οι εντολές αυτές έχουν δύο τελεστές προέλευσης και ένα τελεστή προορισμού. Ισχύουν και εδώ οι παραπάνω θεωρήσεις για το περιεχόμενο των τελεστών.

5.4 Εντολές ελέγχου του προγράμματος (program - control instructions).

Η ομάδα αυτή των εντολών αποτελείται από 17 εντολές, οι οποίες ελέγχουν και επηρεάζουν τη ροή του κώδικα. Η μέθοδος επανάληψης (repeat mode) επιτρέπει την επανάληψη ενός block κώδικα ή μίας γραμμής αυτού. Υποστηρίζουν branching και πράξεις υπό συνθήκη.

5.5 Παράλληλες εντολές.

Μία πολύ σημαντική κατηγορία αποτελεί αυτή των παράλληλων εντολών. Αυτές οι εντολές προσφέρουν:

- παράλληλο loading σε καταχωρητές
- παράλληλες αριθμητικές πράξεις
- αριθμητικές / λογικές πράξεις παράλληλα με εντολή store.

Πριν από τη δεύτερη εντολή κάθε ζεύγους παράλληλων εντολών προηγούνται δύο κάθετες γραμμές (||).

6. Pipeline.

Ένα από τα χαρακτηριστικά της TMS320C3x οικογένειας επεξεργαστών στο οποίο οφείλεται η υψηλή τους απόδοση είναι το pipeline. Η λειτουργία αυτή αφορά στον παραλληλισμό ή την αλληλοκάλυψη των επιπέδων fetch, αποκωδικοποίησης (decode), διαβάσματος (read) και εκτέλεσης (execute) μιας εντολής.

6.1 Η δομή του pipeline

Οι πέντε βασικές μονάδες της δομής του pipeline των 'C3x DSPs είναι οι εξής:

- *Μονάδα Fetch (F)*

Η μονάδα αυτή κάνει fetch στα words της εντολής από τη μνήμη και αυξάνει τον PC.

- *Μονάδα Decode (D)*

Αποκωδικοποιεί το instruction word και παράγει τη διεύθυνσή του. Επίσης, ελέγχει όλες τις αλλαγές στους βοηθητικούς καταχωρητές και στον stack pointer.

- *Μονάδα Read (R)*

Κάνει το read των τελεστών από τη μνήμη, εάν αυτό είναι απαραίτητο.

- *Μονάδα Execute (E)*

Εάν χρειαστεί, διαβάζει τους τελεστές από το register file, υλοποιεί κάθε απαραίτητη λειτουργία και πράξη και γράφει το αποτέλεσμα στο register file. Αν είναι απαραίτητο, γράφει τα αποτελέσματα προηγούμενων πράξεων στη μνήμη.

- *Κανάλι DMA (DMA)*

Διαβάζει από/γράφει στη μνήμη.

Μία βασική εντολή έχει τέσσερα επίπεδα:

- Fetch
- Decode
- Read
- Execute

Το Σχ.1.11 παρουσιάζει τα τέσσερα επίπεδα της δομής του pipeline. Είναι καταχωρημένα ανάλογα με την εντολή και τον κύκλο εκτέλεσης. Η τέλεια επικάλυψη του pipeline συμβαίνει όταν και οι τέσσερις μονάδες λειτουργούν παράλληλα, στον κύκλο (m).

CYCLE	F	D	R	E
m - 3	W	-	-	-
m - 2	X	W	-	-
m - 1	Y	X	W	-
m	Z	Y	X	W
m + 1	-	Z	Y	X
m + 2	-	-	Z	Y
m + 3	-	-	-	Z

D=Decode, E=Execute, F=Fetch, R=Read, W,X,Y,Z=εντολές

Σχ.1.11: Τα τέσσερα επίπεδα του pipeline

Η προτεραιότητα των παραπάνω επιπέδων είναι η:

1. Execute (υψηλότερη)
2. Read
3. Decode
4. Fetch
5. DMA (χαμηλότερη)

Ένα pipeline conflict προκύπτει όταν μία εντολή είναι έτοιμη να περάσει στο αμέσως επόμενο (κατά προτεραιότητα) επίπεδο pipeline, αλλά το επίπεδο αυτό δεν είναι έτοιμο να δεχτεί νέα είσοδο. Σε αυτή την περίπτωση, η χαμηλότερης προτεραιότητας μονάδα περιμένει έως ότου η υψηλότερης προτεραιότητας μονάδα ολοκληρώσει τη λειτουργία που εκτελεί.

Παρόλο που ο DMA controller έχει χαμηλή προτεραιότητα, μπορούν να ελαχιστοποιηθούν τα conflicts με τη CPU, μέσω ειδικών δομών δεδομένων, αφού διαθέτει δικά του buses δεδομένων και διευθύνσεων.

6.2 Τα pipeline conflicts.

Τα pipeline conflicts των 'C3x μπορούν να διαχωριστούν στις ακόλουθες κατηγορίες:

- *Branch conflicts*
Τα Branch conflicts αφορούν στις εντολές ή πράξεις που διαβάζουν και/ή αλλάζουν τον PC.
- *Register conflicts*
Αφορούν σε καθυστερήσεις που μπορεί να συμβούν όταν γίνεται read/write σε registers που χρησιμοποιούνται για παραγωγή διεύθυνσης.
- *Memory conflicts*
Συμβαίνουν όταν τα εσωτερικά μέρη του επεξεργαστή «ανταγωνίζονται» για τους πόρους μνήμης.

7. Εφαρμογές της TMS320 οικογένειας επεξεργαστών.

Η απόδοση σε πραγματικό χρόνο των επεξεργαστών που περιγράφονται στο παρόν κεφάλαι, καθώς και οι προσαρμοστικότητα αυτών τους κάνει εύχρηστους σε ένα πλήθος εφαρμογών, μέρος των οποίων παρουσιάζεται παρακάτω.

- *DSP γενικής χρήσης*
Convolution, Fast Fourier Transform, Παραγωγή κυματομορφών, κ.ά.
- *Graphics/Imaging*
3-D μετασχηματισμοί, Image Transmission/Compression, κ.
- *Voice/Speech*
Voice mail, Αναγνώριση φωνής, Text-to-Speech, κ.ά.
- *Military*
Sonar/Radar Processing, ασφαλείς επικοινωνίες, κ.ά.
- *Τηλεπικοινωνίες*

Κινητή τηλεφωνία, FAX, Video Conferencing, κ.ά.

8. Πλεονεκτήματα των DSPs.

Η παρούσα εφαρμογή αφορά σε επεξεργασία ψηφιακού σήματος καθώς αναφέρεται σε ψηφιακή αναπαράσταση σημάτων και κάνει χρήση ψηφιακού hardware για την ανάλυση, τροποποίηση και «εξαγωγή» πληροφορίας από τα σήματα αυτά.

Υπάρχουν πολλά πλεονεκτήματα για τη χρήση ψηφιακών τεχνικών και, επομένως, ψηφιακών επεξεργαστών σήματος (DSPs). Μερικά από αυτά είναι:

- *Ευκαμψία συστήματος.*

Οι συναρτήσεις ενός DSP συστήματος μπορούν εύκολα να τροποποιηθούν και να αναβαθμιστούν, χρησιμοποιώντας διαφορετικά software modules για το ίδιο hardware.

- *Αξιοπιστία*

Η μνήμη και η λογική ενός DSP δεν εκφυλίζεται με την πάροδο του χρόνου. Οι πόροι που διαθέτει και οι αυξημένες δυνατότητες που παρέχει σε όποιον σχεδιάζει τέτοιου είδους συστήματα υπερτερούν αναμφισβήτητα έναντι άλλων τρόπων υλοποίησης μιας εφαρμογής που επεξεργάζεται ψηφιακό σήμα.

- *Πολυπλοκότητα*

Χρησιμοποιώντας κανείς DSP μπορεί να υλοποιήσει εξεζητημένες εφαρμογές με φορητές και μικρού βάρους συσκευές. Οι DSP αλγόριθμοι μπορούν να αναπτυχθούν, αναλυθούν και εξομοιωθούν χρησιμοποιώντας υψηλού επιπέδου εργαλεία και γλώσσες software.

- *$CPI = 1$*

Καθώς το ρολόι του TMS320C31 λειτουργεί στα 25MHz και η απόδοσή του φτάνει τα 25MIPS, είναι προφανές ότι έχει $CPI = 1$. Το γεγονός αυτό υποδεικνύει ευέλικτο και εύχρηστο σετ εντολών.

Το maximum όριο των 50MFLOPS και των 25MIPS μπορεί να επιτευχθεί με κατάλληλο συνδυασμό εντολών κατά τη δημιουργία κώδικα. Στις παράλληλες εντολές, για παράδειγμα, εάν τα ορίσματα αυτών είναι τέτοια ώστε να μην υπάρχει dependency μεταξύ αυτών, τότε και η απόδοση του κώδικα αυξάνεται.

- *Floating point επεξεργαστής*

Μπορεί να αποθηκεύσει 32-bit data. Η ταχύτητα του είναι περίπου η ίδια με αυτή των fixed point επεξεργαστών, γιατί το hardware αυτών έχει

βελτιστοποιηθεί για μαθηματικές πράξεις. Τα data και registers buses έχουν εύρος 32 bits, έναντι των 16 bits των fixed point επεξεργαστών και ο multiplier όπως και η ALU μπορούν να αποδώσουν γρήγορα floating point πράξεις. Παρέχουν καλύτερη ακρίβεια, ενώ έχουν μικρότερο development cycle, αφού ο προγραμματιστής, γενικά, δεν χρειάζεται να ανησυχεί για overflows/underflows.

Επιπλέον, όσον αφορά στην απόδοση των floating point DSPs, πρέπει να γίνει αναφορά στο signal-to-noise ratio. Στην περίπτωση της αποθήκευσης ενός αριθμού σε 32-bit floating point format, η απόσταση αυτού με τον αμέσως επόμενο αριθμό, το «γείτονά» του είναι περίπου το ένα εκατομμυριοστό του αριθμού. Πρέπει, λοιπόν, να γίνει στρογγυλοποίηση της τιμής στο μισό της απόστασης των δύο αριθμών. Με άλλα λόγια, κάθε φορά που αποθηκεύεται ένας αριθμός σε floating point format, προστίθεται θόρυβος στο σήμα. Το ίδιο συμβαίνει και όταν αυτός αποθηκεύεται ως 16-bit fixed point τιμή. Αυτό που αλλάζει είναι ότι ο θόρυβος σε αυτή την περίπτωση είναι πολύ μεγαλύτερος, αφού η απόσταση, πλέον, μεταξύ δύο γειτονικών αριθμών είναι μεγαλύτερη.

Η TMS320C3x γενιά επεξεργαστών ανήκει στην ευρύτερη κατηγορία αρχιτεκτονικής Harvard. Έχει, δηλαδή, ξεχωριστό χώρο μνήμης για το πρόγραμμα (κώδικα) και για τα δεδομένα, ώστε να υπάρχει η δυνατότητα να γίνεται παράλληλη πρόσβαση σε αυτά.

Τέλος, το software για DSPs αναπτύσσεται, συνήθως, σε γλώσσα C/C++ ή assembly ή ακόμη και συνδυασμό αυτών. Στο παρόν εγχείρημα, προτιμήθηκε η assembly καθώς δίνει τη δυνατότητα πλήρους ελέγχου των συναρτήσεων του επεξεργαστή. Με αυτό τον τρόπο, προκύπτουν αποδοτικά προγράμματα, λόγω της δυνατότητας λεπτομερούς σχεδιασμού, αλλά και ταχύτεροι κώδικες. Υπάρχει ένα προς ένα αντιστοίχιση μεταξύ των εντολών του προγράμματος και των ενεργειών που εκτελεί ο επεξεργαστής. Απαιτεί, βέβαια, πλήρη και βαθιά κατανόηση της αρχιτεκτονικής του DSP.

ΚΕΦΑΛΑΙΟ 2^ο : *ΗΛΕΚΤΡΟΝΙΚΑ ΙΣΧΥΟΣ – SPWM*

1. Εισαγωγή

Ο σκοπός των Ηλεκτρονικών Ισχύος είναι ο έλεγχος της ροής της ισχύος, διαμορφώνοντας κατάλληλα την εκάστοτε παρεχόμενη τάση, χρησιμοποιώντας ημι-αγώγιμες μηχανές ισχύος. Τα πρόσφατα χρόνια, ο τομέας των Ηλεκτρονικών Ισχύος παρουσίασε ιδιαίτερη ανάπτυξη. Αυτό οφείλεται στα πλεονεκτήματα της μικρο-ηλεκτρονικής, τομέας που έκανε την επανάστασή του τα τελευταία χρόνια και που οδήγησε στην ανάπτυξη γραμμικών ολοκληρωμένων κυκλωμάτων και επεξεργαστών ψηφιακού σήματος, όπως είναι οι ελεγκτές (controllers) στα συστήματα ηλεκτρονικών ισχύος.

Τα συστήματα Ηλεκτρονικών Ισχύος αποτελούνται από έναν ή περισσότερους μετατροπείς ηλεκτρονικών ισχύος, οι οποίοι χρησιμοποιούν ημιαγώγιμα στοιχεία ισχύος ελεγχόμενα από ολοκληρωμένα κυκλώματα. Ο μετατροπέας είναι το βασικό στοιχείο αυτών των συστημάτων. Ελέγχει και μορφοποιεί μία είσοδο πλάτους V_i , συχνότητας f_i και αριθμού φάσεων m_i σε έξοδο πλάτους V_o , συχνότητας f_o και αριθμού φάσεων m_o . Η ροή ισχύος στους μετατροπείς μπορεί να είναι αμφίδρομη. Πιο ειδικά, ένας dc-dc μετατροπέας, μετασχηματίζει ένα dc επίπεδο τάσης σε άλλο επίπεδο τάσης. Σε ac-dc μετατροπή, ο μετατροπέας αναφέρεται ως ανορθωτής (rectifier), εάν η ισχύς ρέει από ac σε dc, ενώ, στην αντίθετη περίπτωση, ο μετατροπέας ονομάζεται αντιστροφέας (inverter).

Υπάρχουν πολλοί τρόποι για να ταξινομηθούν οι μετατροπείς ηλεκτρονικών ισχύος. Θα μπορούσε να γίνει ταξινόμηση βάση του τύπου της συσκευής που χρησιμοποιείται, της λειτουργίας του μετατροπέα, κ.ά. Δυστυχώς, τα παραπάνω κριτήρια δεν παρέχουν καλά-ορισμένες κατηγορίες. Ένας πιο ασφαλές τρόπος ταξινόμησής τους είναι η κατηγοριοποίηση των μετατροπέων με βάση τον τρόπο με τον οποίο τα στοιχεία του εναλλάσσονται. Έτσι έχουμε τους:

- ο *Μετατροπείς συχνότητας*, όπου η τάση που είναι παρούσα στη μία πλευρά του μετατροπέα διευκολύνει τη διαδικασία του turn-off των ημιαγώγιμων στοιχείων του. Τα στοιχεία αυτά ενεργοποιούνται (turn-on) κλειδώνοντας σε φάση στην γραμμική τάση. Γι' αυτό το λόγο, τα στοιχεία του μετατροπέα εναλλάσσονται μεταξύ on και off σε μια γραμμική συχνότητα των 50 ή 60 Hz.

- ο *Μετατροπείς με χρήση διακοπών*, όπου οι ελεγχόμενοι διακόπτες ανοίγουν και κλείνουν σε συχνότητες αρκετά υψηλές συγκριτικά με τη βασική συχνότητα. Παρ'όλα αυτά, η έξοδος του μετατροπέα μπορεί να είναι είτε dc ή σε μία συχνότητα συγκρίσιμη με αυτή της εισόδου.

Οι παραπάνω ταξινομήσεις μπορούν να χωριστούν περαιτέρω σε υπο-κατηγορίες, ανάλογα με τον τύπο της μετατροπής, για παράδειγμα, dc-dc μετατροπή, ανόρθωση, αντιστροφή, κ.ά, κάτι που θα μελετηθεί συνοπτικά παρακάτω.

2. Κατηγορίες μετατροπέων (converters).

Στην παρούσα ενότητα, θα γίνει σύντομη αναφορά σε μερικούς μετατροπείς, στον τρόπο λειτουργίας τους και τη χρησιμότητά τους.

Οι **dc-dc μετατροπείς** χρησιμοποιούνται ευρέως σε dc-motor drive εφαρμογές. Συχνά, η είσοδος σε αυτούς τους μετατροπείς είναι μία μη-ρυθμισμένη dc τάση, η οποία μπορεί να διακυμανθεί εξαιτίας των μεταβολών της έντασης της τάσης. Χρησιμοποιούνται για τη μετατροπή της παραπάνω αναφερθείσας εισόδου σε μία ελεγχόμενη dc έξοδο επιθυμητού επιπέδου τάσης.

Στις περισσότερες εφαρμογές ηλεκτρονικών ισχύος, η είσοδος είναι της μορφής μιας ac ημιτονοειδούς τάσης, στα 50 Hz ή στα 60 Hz, η οποία μετατρέπεται σε dc τάση. Η μετατροπή αυτή με ένα μη-ελεγχόμενο τρόπο γίνεται χρησιμοποιώντας **ανορθωτές με διόδους από 60 Hz ac σε μη-ελεγχόμενο dc**. Σε αυτούς η ροή της ισχύος μπορεί να γίνεται μόνο από την ac στην dc πλευρά του συστήματος. Στην πλειονότητά τους, τα ηλεκτρονικά ισχύος χρησιμοποιούν τέτοιους ανορθωτές. Στις περισσότερες εφαρμογές, αυτοί τροφοδοτούνται κατευθείαν, χωρίς την παρουσία μετασχηματιστή των 60 Hz. Σε πιο μοντέρνα συστήματα, η αποφυγή χρήσης μετασχηματιστή είναι πολύ σημαντική.

Υπάρχουν, όμως και εφαρμογές, όπως οι ac ή dc motor drives, όπου είναι απαραίτητο να μπορεί να είναι ελεγχόμενη η dc τάση εξόδου. Αυτό επιτυγχάνεται με **ανορθωτές και αντιστροφείς ελεγχόμενης φάσης**. Στο παρελθόν, τέτοιοι μετατροπείς, χρησιμοποιούνταν σε ένα μεγάλο αριθμό εφαρμογών για να ελέγχουν τη ροή της ηλεκτρικής ισχύος. Πλέον, χρησιμοποιούνται κυρίως σε τριφασικές εφαρμογές, υψηλής ισχύος, λόγω της διαθεσιμότητας ελεγχόμενων διακοπών για υψηλές τιμές τάσης και ρεύματος. Κι αυτό συμβαίνει καθώς, στα παραπάνω συστήματα είναι απαραίτητο ή επιθυμητό να μπορεί να υπάρχει έλεγχος της ροής της ισχύος είτε από ac σε dc ή το αντίστροφο. Παραδείγματα τέτοιων εφαρμογών είναι μετατροπείς σε συστήματα μεταφοράς dc ισχύος υψηλής τάσης.

Τέλος, υπάρχουν και οι **dc-ac αντιστροφείς**, με τους οποίους ασχολείται διεξοδικά η παρούσα εργασία. Χρησιμοποιούνται ευρέως σε ac-motor drives και σε μη-διακοπόμενους παροχείς ac ισχύος, όπου το αντικείμενο είναι η παραγωγή ac εξόδου, της οποίας το πλάτος και η συχνότητα να μπορεί να ελέγχεται. Για παράδειγμα, σε ένα ac φορτίο κινητήρα, η έξοδος είναι επιθυμητό να είναι ημιτονοειδής και να μπορεί να προσαρμοστεί σε επίπεδα τάσης και συχνότητας. Αυτό επιτυγχάνεται με έναν dc-ac αντιστροφέα, ο οποίος δέχεται ως είσοδο μία dc τάση και παράγει την επιθυμητή ac τάση εξόδου. Στην επόμενη ενότητα, θα γίνει διεξοδική αναφορά στους dc-ac αντιστροφείς.

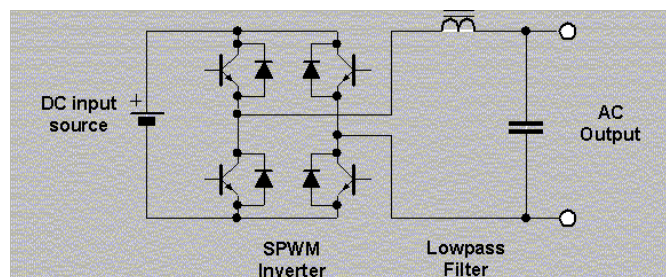
3. DC –AC Αντιστροφείς: $dc \Leftrightarrow$ Ημιτονοειδές ac.

Οι αντιστροφείς που περιγράφονται στην παρούσα ενότητα είναι μετατροπείς όπου η ροή της ισχύος είναι αμφίδρομης κατεύθυνσης. Παρακάτω, θα γίνει αναφορά σε αντιστροφείς μονοφασικής και τριφασικής ac εξόδου. Η είσοδος αυτών θα θεωρηθεί ότι είναι μία dc πηγή τάσης. Τέτοιου είδους αντιστροφείς καλούνται αντιστροφείς πηγής τάσης (voltage source inverters – VSIs).

Οι VSIs χωρίζονται σε τρεις γενικές κατηγορίες:

- *Αντιστροφείς διαμόρφωσης εύρους παλμών (Pulse-width-modulated Inverters).*

Η dc τάση εισόδου είναι απαραίτητα σταθερή σε πλάτος. Έτσι, ο αντιστροφέας πρέπει να ελέγχει το πλάτος και τη συχνότητα της ac εξόδου. Αυτό επιτυγχάνεται εφαρμόζοντας κανείς PWM στους διακόπτες του, και γι'αυτό το λόγο ονομάζονται και **PWM Inverters**. Υπάρχουν διάφοροι τρόποι ώστε να γίνει η διαμόρφωση εύρους παλμών με στόχο η τάση εξόδου να μοιάζει όσο περισσότερο γίνεται με ημίτονο. Ένας από αυτούς είναι και η Ημιτονοειδής διαμόρφωση εύρους παλμών (sinusoidal PWM), η οποία και θα αναλυθεί λεπτομερώς παρακάτω, καθώς αποτελεί το αντικείμενο της εργασίας αυτής.



Σχ. 2.1: Αντιστροφέας τύπου γέφυρας (bridge-type inverter)

Στο παραπάνω σχήμα (Σχ.2.1), παρουσιάζεται η πιο απλή μορφή DC/AC Inverter, η τύπου γέφυρας μορφή (bridge-type), όπου μία γέφυρα ελέγχεται βάση της SPWM. Η SPWM κυματομορφή που προκύπτει φιλτράρεται για να παράγει την επιθυμητή έξοδο. Το μειονέκτημα της μεθόδου αυτής είναι πως στην περίπτωση που η τάση εισόδου είναι χαμηλή, απαιτείται μετασχηματιστής ισχύος μεγάλου μεγέθους, βάρους και υψηλού κόστους.

- ο *Αντιστροφείς τετραγωνικής κυματομορφής (Square-wave Inverters).*

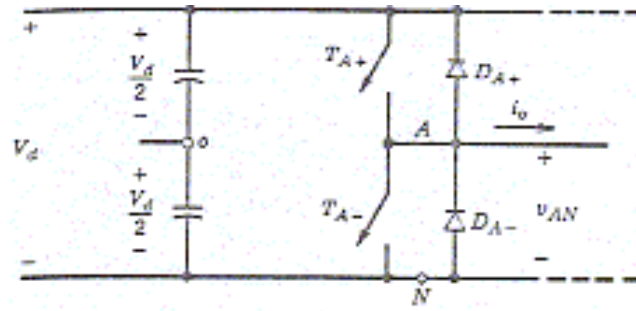
Σε αυτούς τους αντιστροφείς, η dc τάση εισόδου ελέγχεται με σκοπό να ρυθμίζει το πλάτος της ac εξόδου, και έτσι, ο αντιστροφέας πρέπει να ελέγχει μόνο τη συχνότητα εξόδου. Η τάση εξόδου έχει τη μορφή τετραγωνικού παλμού, εξ'ού και το όνομα των αντιστροφέων αυτών.

- ο *Μονοφασικοί Αντιστροφείς με ακύρωση τάσης (Single-phase Inverters with voltage cancellation).*

Πρόκειται για αντιστροφείς με μονοφασική έξοδο, όπου είναι πιθανό να ελέγχεται το πλάτος και η συχνότητα της εξόδου, με την dc είσοδο σταθερή και τους διακόπτες του αντιστροφέα μη-διαμορφωμένους κατά εύρος παλμών. Σε αυτή την περίπτωση, η κυματομορφή εξόδου θα είναι τετραγωνική. Σημειώνεται ότι αυτή η τεχνική δεν λειτουργεί με τριφασικούς αντιστροφείς.

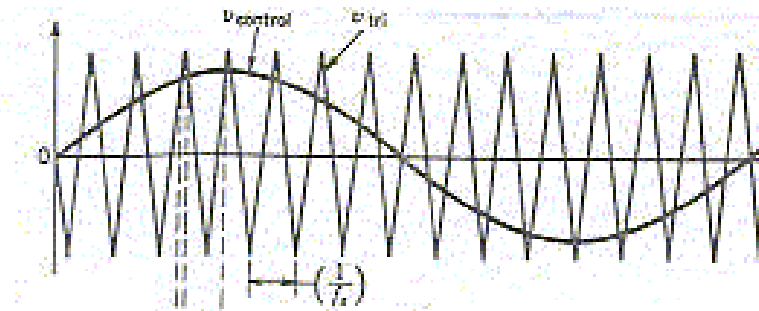
4. Αντιστροφείς διαμόρφωσης εύρους παλμών.

Η εφαρμογή PWM σε κυκλώματα αντιστροφέων, έχει ως στόχο μια ημιτονοειδή έξοδο με ελεγχόμενο πλάτος και συχνότητα. Για την παραγωγή ημιτονοειδούς εξόδου σε επιθυμητή συχνότητα, γίνεται σύγκριση ενός ημιτονοειδούς σήματος ελέγχου της συχνότητας που θέλουμε με μία τριγωνική κυματομορφή. Η συχνότητα του τριγώνου καθορίζει τη συχνότητα των διακοπών του inverter και διατηρείται σταθερή, όπως σταθερό διατηρείται και το πλάτος του τριγώνου, V_{tri} . Δείγμα του αντιστροφέα ενός σκέλους (one-leg inverter) στον οποίο αναφέρονται τα παραπάνω, παρουσιάζεται στο Σχ.2.2.



Σχ.2.2: One-leg switch-mode inverter

Ξεκινώντας την ανάλυση της διαμόρφωσης εύρους παλμών, ας οριστεί ότι η τριγωνική κυματομορφή V_{tri} έχει συχνότητα f_s , η οποία καθορίζει τη συχνότητα με την οποία ανοιγο-κλείνουν οι διακόπτες του αντιστροφέα. Ονομάζεται και *συχνότητα φορέα*, f_s . Το σήμα ελέγχου $V_{control}$ χρησιμοποιείται για τη διαμόρφωση του switch duty ratio και έχει συχνότητα f_1 , η οποία είναι η επιθυμητή βασική συχνότητα της εξόδου τάσης του inverter. Καλείται και *συχνότητα διαμόρφωσης*, f_1 . Ο συσχετισμός σήματος ελέγχου – σήματος φορέα φαίνεται στο Σχ.2.3. Θεωρείται, δε, σίγουρο, ότι η έξοδος του inverter δεν θα είναι ένα τέλειο ημίτονο, αλλά θα περιέχει συνιστώσες τάσης σε αρμονικές συχνότητες της f_1 .



Σχ.2.3: Διαμόρφωση εύρους παλμών

Ο συντελεστής διαμόρφωσης πλάτους m_a , ορίζεται ως:

$$m_a = \frac{V_{control}}{V_{tri}}, \quad (\Sigma\chi.1)$$

όπου $V_{control}$ είναι η peak τιμή του πλάτους του σήματος ελέγχου. Το πλάτος του τριγωνικού σήματος, V_{tri} , διατηρείται σταθερό.

Ο συντελεστής διαμόρφωσης συχνότητας m_f , ορίζεται ως:

$$m_f = \frac{f_s}{f_l}. \quad (\Sigma\chi.2)$$

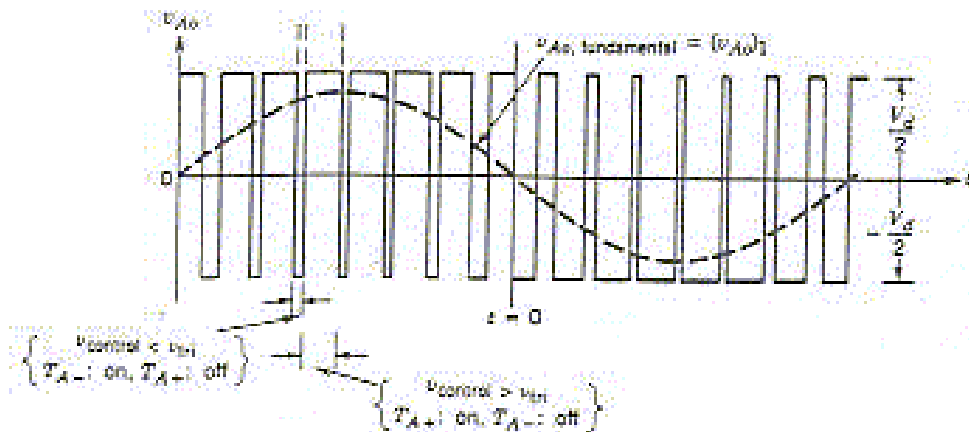
Οι διακόπτες T_{A+} και T_{A-} ελέγχονται βάσει της σύγκρισης του $V_{control}$ με το V_{tri} και η τάση εξόδου προκύπτει ως εξής:

$$V_{control} > V_{tri} \, , \qquad T_{A+} \text{ είναι κλειστός}, \qquad V_{Ao} = \frac{1}{2} V_d$$

(Σχ.3)

$$V_{control} < V_{tri}, \quad T_{A-} \text{ είναι κλειστός,} \quad V_{Ao} = -\frac{1}{2}V_d.$$

Το γεγονός ότι δεν μπορεί να είναι και οι δύο διακόπτες κλειστοί την ίδια στιγμή, οδηγεί στο συμπέρασμα ότι η τάση εξόδου V_{Ao} κυμαίνεται μεταξύ δύο τιμών, της $\frac{1}{2}V_d$ και της $-\frac{1}{2}V_d$, πράγμα που φαίνεται και στο Σχ.2.4.



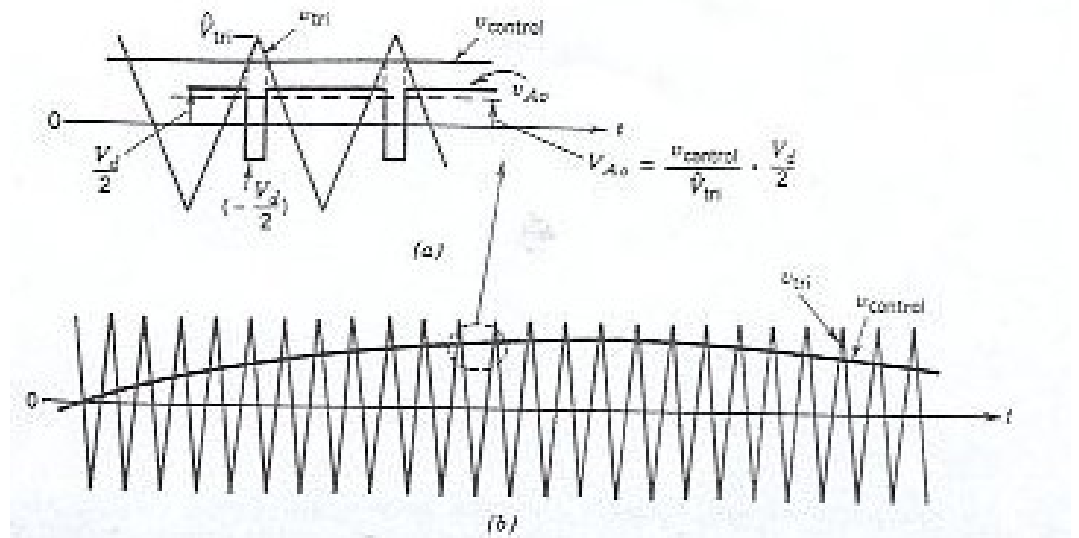
Σχ. 2.4: Παλμοσειρά εξόδου PWM

Στο παραπάνω σχήμα, η τάση εξόδου έχει σχεδιαστεί για $m_a = 0.8$ και $m_f = 15$. Από αυτό προκύπτουν κάποιες σημαντικές παρατηρήσεις (για $m_a \leq 1.0$):

- Το πλάτος της βασικής αρμονικής εξόδου ισούται με m_a φορές την τιμή $\frac{1}{2}V_d$,

$$\text{δηλαδή:} \quad V_{Ao} = \frac{V_{control}}{V_{tri}} \frac{V_d}{2}, \quad V_{control} \leq V_{tri} . \quad (\Sigma\chi.4)$$

Υποθέτουμε, αν και δεν είναι αναγκαίο, ότι η $V_{control}$ αλλάζει πολύ λίγο μέσα σε μία περίοδο, δηλαδή ότι ο m_f είναι πολύ μεγάλος. Έτσι, όπως δείχνει και το Σχ.2.5, για μία περίοδο θεωρούμε πως η $V_{control}$ είναι σταθερή.



Σχ.2.5: Ημιτονοειδές PWM

- Οι αρμονικές της τάσης εξόδου του αντιστροφέα εμφανίζονται ως πλευρικές ζώνες συχνοτήτων γύρω από την βασική συχνότητα και τα παράγωγά της, δηλαδή, γύρω από τις αρμονικές m_f , $2m_f$, $3m_f$, $4m_f$, κ.λ.π. Αυτό ισχύει για όλες τις τιμές του εύρους τιμών 0-1 του συντελεστή m_a .

Όταν $m_f \leq 9$, πράγμα που συνήθως ισχύει, εκτός από περιπτώσεις πολύ υψηλής ισχύος, τα πλάτη των αρμονικών είναι σχεδόν ανεξάρτητα από τον m_f , παρόλο που αυτός ο συντελεστής καθορίζει τις συχνότητες στις οποίες προκύπτουν αρμονικές. Θεωρητικά, οι συχνότητες όπου παρουσιάζονται αρμονικές, προκύπτουν από:

$$f_h = (jm_f \pm k)f_1,$$

όπου, η αρμονική h αντιστοιχεί στην k πλευρική του j^* m_f :

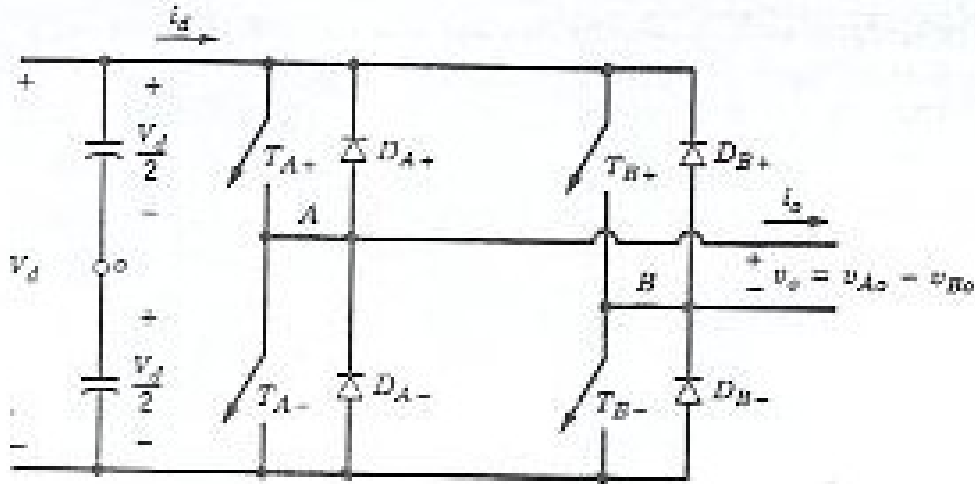
$$h = j(m_f) \pm k, \quad (\Sigma\chi.5)$$

με $h=1$ για την βασική συχνότητα. Για περιττές τιμές του j , οι αρμονικές υπάρχουν μόνο για άρτιες τιμές του k . Για άρτιες τιμές του j , οι αρμονικές υπάρχουν μόνο για περιττές τιμές του k .

4.1. Αντιστροφείς πλήρους γέφυρας (Full-bridge inverters).

Οι αντιστροφείς πλήρους γέφυρας αποτελούνται από δύο one-leg αντιστροφείς, τύπου του Σχ.2.2. Με την ίδια dc τάση εισόδου, η μέγιστη τάση εξόδου του αντιστροφέα είναι δύο

φορές η τάση εξόδου του one-leg αντιστροφέα. Αυτό δηλώνει πως για την ίδια ισχύ, το ρεύμα εξόδου και τα ρεύματα των διακοπών είναι τα μισά από αυτά του half-bridge αντιστροφέα. Σε υψηλά επίπεδα ισχύος αυτό αποτελεί πλεονέκτημα.



Σχ.2.6: Μονοφασικός αντιστροφέας πλήρους γέφυρας

4.1.1. Διαμόρφωση PWM με διπολική τάση εξόδου (Bipolar Voltage Switching).

Στο παραπάνω σχήμα, οι διαγώνια αντίθετοι διακόπτες (T_{A+}, T_{B-}) και (T_{A-}, T_{B+}) ανοιγοκλείνουν ως ζεύγη διακοπών, τα ζεύγη 1 και 2 αντίστοιχα. Ακολουθώντας αυτόν τον τρόπο λειτουργίας των διακοπών, η τάση εξόδου του μέρους A του αντιστροφέα είναι ίδια με την έξοδο του one-leg αντιστροφέα, του οποίου η λειτουργία περιγράφηκε στην προηγούμενη ενότητα, και καθορίζεται με τον ίδιο τρόπο, δηλαδή με τη σύγκριση των $V_{control}$ και V_{tri} . Η έξοδος του σκέλους B του αντιστροφέα πλήρους γέφυρας είναι αντίθετη του σκέλους A του αντιστροφέα. Για παράδειγμα, όταν ο διακόπτης T_{A+} είναι κλειστός, δηλαδή είναι σε on κατάσταση, και το V_{Ao} είναι ίσο με $\frac{1}{2}V_d$, τότε και ο T_{B-} είναι κλειστός και

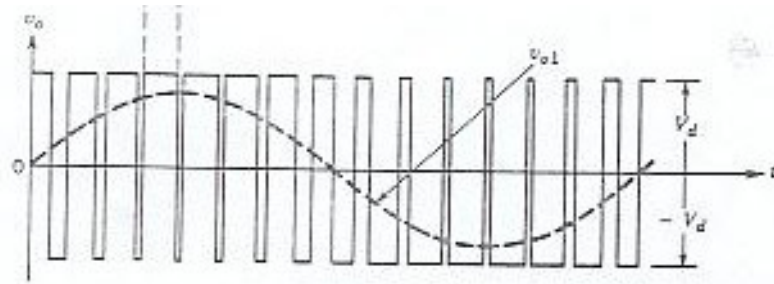
ισχύει: $V_{Bo} = -\frac{1}{2}V_d$. Έτσι,

$$V_{Bo}(t) = -V_{Ao}(t) \quad (\Sigma\chi.4)$$

και

$$V_o(t) = V_{Ao}(t) - V_{Bo}(t) = 2V_{Ao}(t) \quad (\Sigma\chi.5)$$

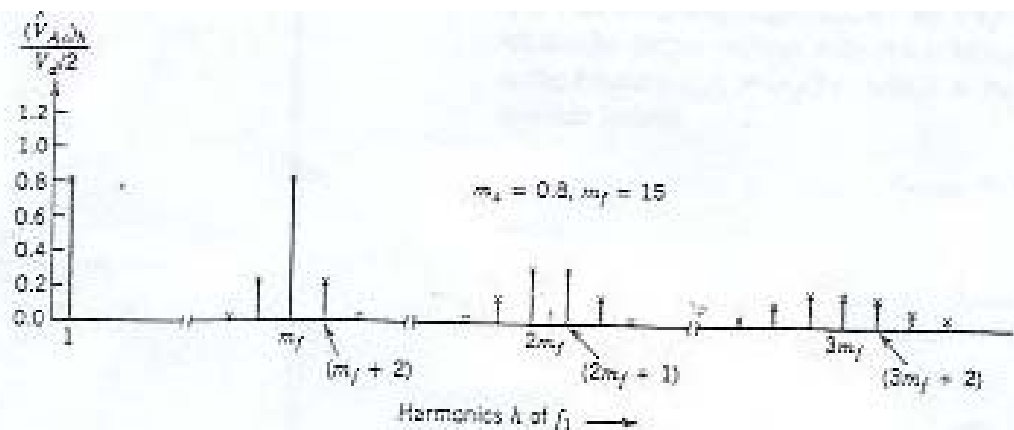
Η κυματομορφή εξόδου παρουσιάζεται στο Σχ.2.7



Σχ.2.7: PWM με διπολική τάση εξόδου

Παρατηρούμε ότι η έξοδος κυμαίνεται μεταξύ των τιμών V_d και $-V_d$.

Οι αρμονικές της τάσης εξόδου του αντιστροφέα εμφανίζονται ως πλευρικές ζώνες, με κέντρο τη συχνότητα μετάβασης και τις πολλαπλάσιές της, δηλαδή γύρω από τις αρμονικές m_f , $2m_f$, $3m_f$, κ.ο.κ., όπως φαίνεται και στο φάσμα της τάσης εξόδου στο Σχ.2.8.



Σχ.2.8: Φάσμα αρμονικών

Η αρμονική τάξη h μπορεί να γραφεί ως:

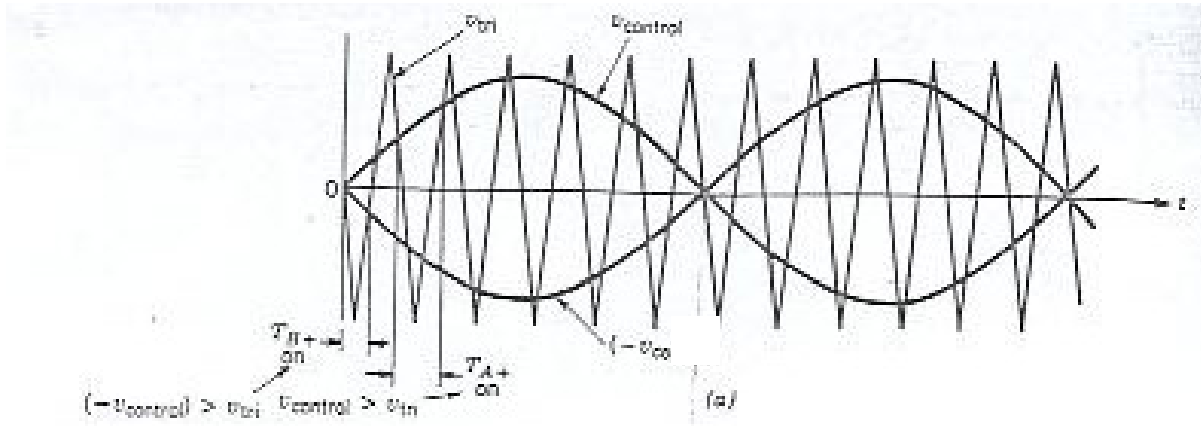
$$h = j(m_f) \pm k, \quad (\Sigma\chi.6)$$

όπου η θεμελιώδης συχνότητα αντιστοιχεί στο $h = 1$. Για περιττές τιμές του j , οι αρμονικές υπάρχουν μόνο για άρτιες τιμές του k . Για άρτιες τιμές του j , οι αρμονικές υπάρχουν μόνο για περιττές τιμές του k .

4.1.2. Διαμόρφωση PWM με μονοπολική τάση εξόδου (Unipolar Voltage Switching).

Σε PWM με εναλλαγή των τιμών της τάσης εξόδου μεταξύ του μηδενός και μίας τιμής, οι διακόπτες στα δύο σκέλη του αντιστροφέα πλήρους γέφυρας του Σχ.2.5, δεν

ανοιγοκλείνουν ταυτόχρονα, όπως στην προηγούμενη ενότητα. Πλέον, τα σκέλη Α και Β ελέγχονται χωριστά, κάνοντας σύγκριση του V_{tri} με το $V_{control}$ και το $-V_{control}$, αντίστοιχα.



Σχ. 2.9: PWM με μονοπολική τάση εξόδου

Όπως φαίνεται και στο Σχ.2.9, η σύγκριση του $V_{control}$ με την τριγωνική κυματομορφή παράγει τα ακόλουθα για τον έλεγχο των διακοπών του μέρους Α του αντιστροφέα:

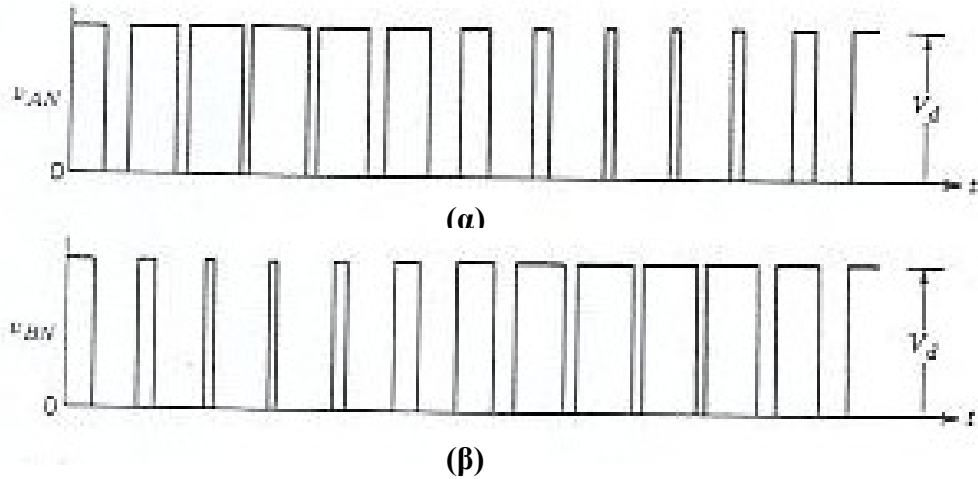
$$V_{control} > V_{tri}, \quad T_{A+} \text{ είναι κλειστός,} \quad V_{AN} = V_d \quad (\Sigma\chi.7)$$

$$V_{control} < V_{tri}, \quad T_{A-} \text{ είναι κλειστός,} \quad V_{AN} = 0.$$

Στο Σχ.2.10(α) φαίνεται η τάση εξόδου του σκέλους Α του αντιστροφέα, ενώ στο Σχ.3.10(β) η τάση εξόδου του σκέλους Β. Για τον έλεγχο των διακοπών του μέρους Β, συγκρίνεται η $-V_{control}$ με την ίδια, με παραπάνω, τριγωνική κυματομορφή, και προκύπτει:

$$-V_{control} > V_{tri}, \quad T_{B+} \text{ είναι κλειστός,} \quad V_{BN} = V_d \quad (\Sigma\chi.8)$$

$$-V_{control} < V_{tri}, \quad T_{B-} \text{ είναι κλειστός,} \quad V_{BN} = 0.$$



Σχ.2.10: Επιμέρους τάσεις εξόδου του inverter

Από τα Σχ.2.9 και Σχ.2.10, παρατηρεί κανείς πως υπάρχουν τέσσερις συνδυασμοί διακοπών σε οη κατάσταση, οι ακόλουθοι:

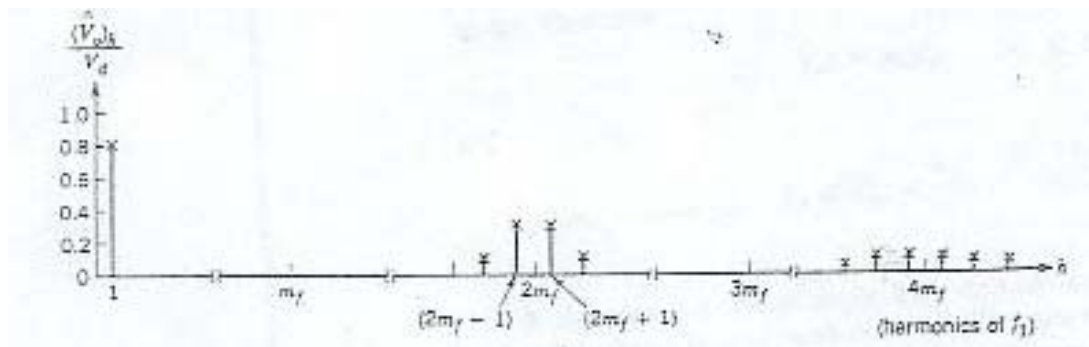
1. T_{A+}, T_{B-} on: $V_{AN} = V_d, V_{BN} = 0, V_o = V_d$
2. T_{A-}, T_{B+} on: $V_{AN} = 0, V_{BN} = V_d, V_o = -V_d$
3. T_{A+}, T_{B+} on: $V_{AN} = V_d, V_{BN} = V_d, V_o = 0$
4. T_{A-}, T_{B-} on: $V_{AN} = 0, V_{BN} = 0, V_o = 0$ (Σχ.9)

Παρατηρούμε πως, όταν συμβαίνει μία αλλαγή στην κατάσταση των διακοπών, η τάση εξόδου εναλλάσσεται μεταξύ των επιπέδων τάσης 0 και $+V_d$, ή μεταξύ των 0 και $-V_d$. Γι' αυτό το λόγο και το παρόν PWM σχήμα ονομάζεται PWM με εναλλαγή τάσης ενός πόλου, σε αντίθεση με το διπολικό PWM σχήμα, όπου η τάση εναλλασσόταν μεταξύ $+V_d$ και $-V_d$. Το πλεονέκτημα του PWM σχήματος που περιγράφεται εδώ είναι το γεγονός ότι η τάση εξόδου μπορεί να «μεταπηδήσει» κατά V_d σε αλλαγή της κατάστασης των διακοπών του αντιστροφέα, ενώ στο διπολικό σχήμα κατά $2V_d$. Αξιοσημείωτο είναι επίσης το ότι έχει διπλασιαστεί η συχνότητα εναλλαγής των διακοπών, πράγμα σημαντικό όσον αφορά στις αρμονικές της εξόδου.

Η αρμονική τάξη h μπορεί να γραφεί ως:

$$h = j(2m_f) \pm k, \quad (\text{Σχ.10})$$

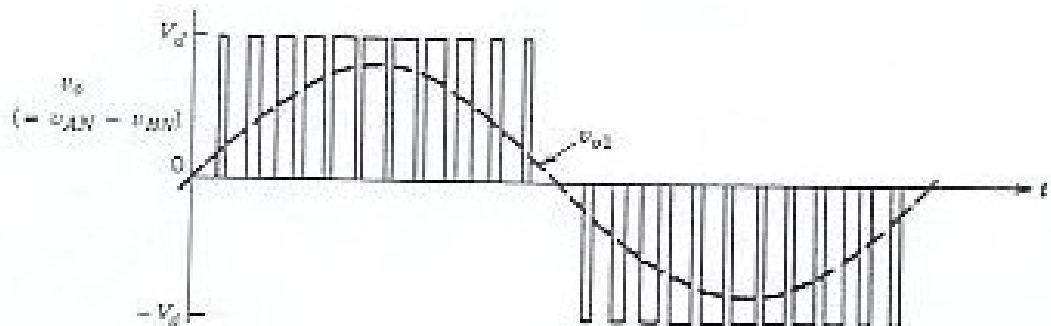
όπου οι αρμονικές υπάρχουν ως πλευρικές ζώνες γύρω από τη $2m_f$ και τις πολλαπλάσιες της $2m_f$. Εφόσον το h παίρνει μόνο περιττές τιμές και το k παίρνει μόνο περιττές τιμές.



Σχ.2.11: Φάσμα αρμονικών

Το πλεονέκτημα του διπλασιασμού της συχνότητας μετάβασης εμφανίζεται στο φάσμα της τάσης εξόδου (Σχ.2.11), όπου οι χαμηλότερες αρμονικές εμφανίζονται ως πλευρικές ζώνες σε συχνότητα διπλάσια της θεμελιώδους. Σε σύγκριση με τη διπολική μετάβαση τάσης, στη μονοπολική, οι αρμονικές γύρω από το m_f εξαφανίζονται, γεγονός που καταλήγει σε ένα φάσμα χαμηλότερου περιεχομένου.

Η έξοδος, λοιπόν, του παρόντος PWM σχήματος είναι η παρακάτω:



Σχ.2.12: Έξοδος του inverter

Η ανάλυση, έως τώρα, έδωσε βάρος στους μονοφασικούς αντιστροφείς. Δεν αλλάζει στην ουσία της, όταν αναφέρεται σε τριφασικούς αντιστροφείς. Σε αυτή την περίπτωση, οι τρεις διαφορετικές φάσεις έχουν διαφορά 120° μεταξύ τους.

5. Ημιτονοειδής διαμόρφωση εύρους παλμών (SPWM) με χρήση επεξεργαστή DSP (TMS320C31)

5.1 Εισαγωγή

Στη σημερινή εποχή, σε συστήματα ηλεκτρονικών ισχύος, είναι πολύ διαδεδομένη η χρήση DC/AC αντιστροφέων. Πιο συγκεκριμένα, είναι ευρεία η ανάγκη μετατροπής συνεχούς τάσης σε εναλλασσόμενη. Χρησιμοποιούνται σε συστήματα συνεχούς λειτουργίας

(UPS), σε συστήματα ελέγχου μοτέρ (AC motor drives), κ.ά. Σε πολλές εφαρμογές είναι σημαντικό, ο αντιστροφέας να έχει μικρό βάρος και μέγεθος.

Αρχικά χρησιμοποιείται ένας αντιστροφέας ο οποίος παράγει ένα διαμορφωμένο SPWM σήμα στα 50Hz. Στη συνέχεια, ένα χαμηλοπερατό φίλτρο (Lowpass Filter) παράγει ημιτονοειδές σήμα υψηλής τάσης και χαμηλής παραμόρφωσης στα 50Hz. Ο ρυθμός εναλλαγής των διακοπών στη γέφυρα αναστροφής πολικότητας καθορίζεται από τη χαμηλή συχνότητα του σήματος εξόδου (50Hz), έτσι ώστε να έχει μικρές απώλειες και χαμηλό κόστος.

5.1.1 Γενική περιγραφή υλοποίησης της SPWM στον 'C31.

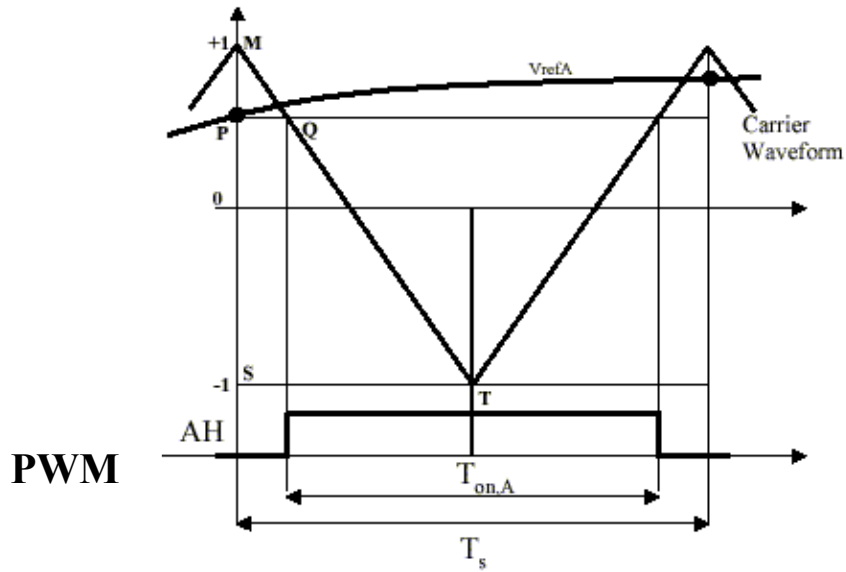
Στα ηλεκτρονικά ισχύος, και ειδικότερα, στον τομέα των κινητήρων, η σχεδίαση είναι τέτοια ώστε να λειτουργούν με ημιτονοειδείς τάσεις εισόδου. Γι' αυτό το λόγο και η έξοδος του αντιστροφέα πρέπει να μοιάζει όσο γίνεται πιο πολύ με ημίτονο. Αυτά τα σήματα εξόδου μεταβαλλόμενου πλάτους και συχνότητας μπορούν να παραχθούν συγκρίνοντάς τα με ένα τριγωνικό φέρον σήμα (carrier) υψηλής συχνότητας στην επιθυμητή συχνότητα.

Με τη βοήθεια του επεξεργαστή TMS320C31, γίνεται δειγματοληψία στο σήμα αναφοράς με ρυθμό ίσο με τη συχνότητα εναλλαγής των διακοπών του αντιστροφέα (switching frequency), τη συχνότητα, δηλαδή, του φορέα. Οι τιμές που προκύπτουν από τη διαδικασία αυτή, αποθηκεύονται σε έναν lookup table, ώστε κάθε φορά να είναι εκ των προτέρων γνωστή η τιμή του ημιτόνου με βάση το οποίο θα γίνει η σύγκριση με το τρίγωνο. Έτσι, η παραγωγή των PWM παλμών γίνεται σαφώς πιο γρήγορα από ότι αν υπολογίζαμε την επόμενη τιμή του ημιτόνου για τη σύγκριση τη στιγμή που τη χρειαζόμασταν.

Η παραγωγή του PWM σήματος γίνεται χρησιμοποιώντας τους δύο timers του TMS320C31. Παράλληλα με αυτούς, λειτουργούν και κάποια flags, που έχουμε ορίσει στο σύστημα, ώστε να είναι γνωστό ανά πάσα στιγμή το πρόσημο των σημάτων μας, και έτσι να γίνει σωστά η όλη διαδικασία.

5.1.2 Εναλλακτικός τρόπος υπολογισμού του Duty Cycle της μεθόδου.

Με βάση το Σχ.2.13, μπορεί κανείς να υπολογίσει το duty cycle του συστήματος.



Σχ.2.13: Παραγωγή του PWM σήματος σε μία περίοδο μετάβασης

Όπως φαίνεται, τα δύο τρίγωνα MPQ και MST είναι όμοια. Συνεπώς,

$$\frac{|PQ|}{|ST|} = \frac{|PM|}{|SM|} \quad (\text{Σχ.11})$$

οπότε:

$$\frac{\frac{1}{2}(T_s - T_{on,A})}{\frac{1}{2}T_s} = \frac{1 - V_{ref,A}}{2} \quad (\text{Σχ.12})$$

Εάν αναλυθεί η Σχέση (12) ως προς $T_{on,A}$, προκύπτει:

$$T_{on,A} = \frac{T_s}{2}(1 + V_{ref,A}) \quad (\text{Σχ.13})$$

Η Σχέση (13) παρέχει κάθε στιγμή το θετικό κύκλο επί του συνόλου του παλμού.

Από όλα τα παραπάνω προκύπτει ότι οι επιθυμητές τάσεις εξόδου προκύπτουν μεταβάλλοντας τη συχνότητα και το πλάτος του υπό διαμόρφωση σήματος. Τα αποτελέσματα των συγκρίσεων των ημιτονοειδών αναφορών με το φέρων σήμα είναι τα PWM σήματα που χρησιμοποιούνται για τον έλεγχο του αντιστροφέα.

ΚΕΦΑΛΑΙΟ 3^ο : *ΥΛΟΠΟΙΗΣΗ ΤΟΥ SPWM ΣΤΟΝ TMS320C31.*

1. Εισαγωγή

Όπως αναφέρθηκε, στην παρούσα εφαρμογή υλοποιήθηκε η μέθοδος PWM με μονοπολική τάση εξόδου (PWM with Unipolar Voltage Switching) στο TMS320C31 DSP. Η υλοποίηση της μεθόδου έγινε σε γλώσσα assembly. Υπάρχει η δυνατότητα να γράψει κανείς κώδικα σε γλώσσα C για την TMS320C3x οικογένεια επεξεργαστών. Παρόλο, όμως, που η C είναι περισσότερο εύχρηστη και εύκολη στη διαχείρισή της, ένας κώδικας γραμμένος σε C δεν μπορεί να συγκριθεί στην αποδοτικότητα και την ταχύτητα ενός κώδικα γραμμένου σε assembly, για λόγους οι οποίοι έχουν ήδη αναφερθεί στο κεφάλαιο που αφορά στους DSP. Γι'αυτό το λόγο η υλοποίηση της μεθόδου SPWM στον επεξεργαστή TMS320C31 έγινε χρησιμοποιώντας γλώσσα assembly. Το γεγονός αυτό, φυσικά, βοήθησε στην καλύτερη και βαθύτερη κατανόηση της αρχιτεκτονικής των συγκεκριμένων επεξεργαστών.

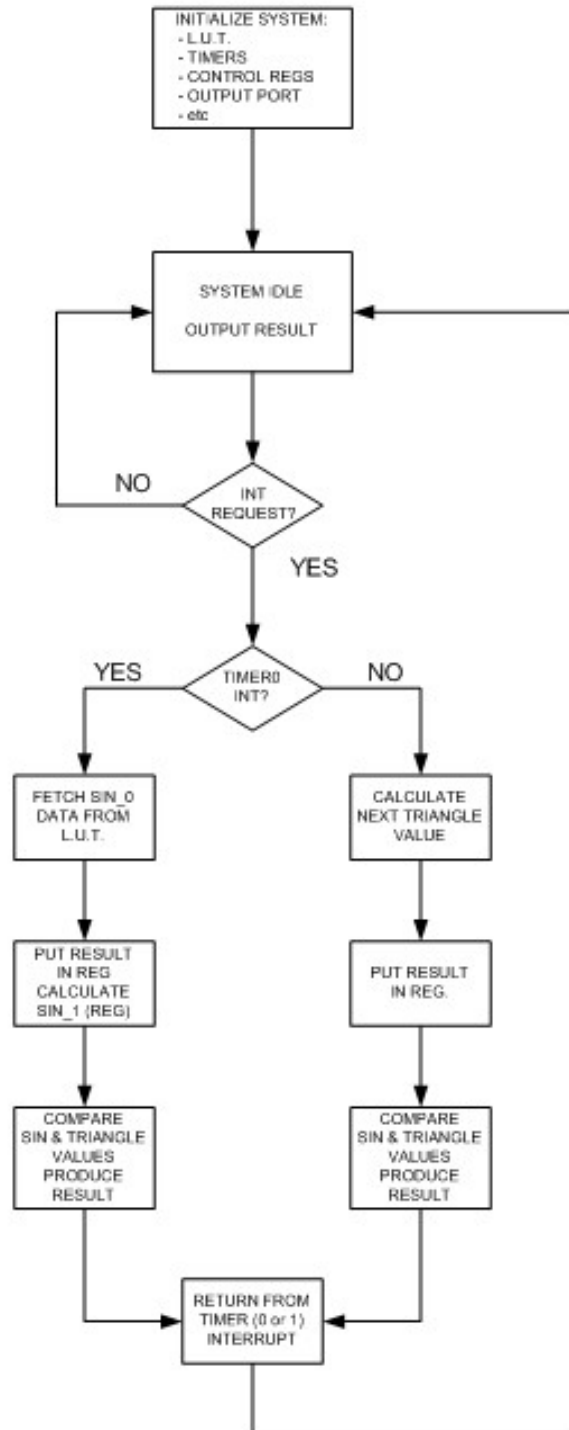
Αξίζει να αναφερθεί πως η υλοποίηση της μεθόδου PWM στηρίχθηκε κυρίως στους δύο timers, καθώς και την εσωτερική μνήμη, πόροι διαθέσιμοι του TMS320C31. Επιπλέον, οι καταχωρητές που χρησιμοποιήθηκαν, είναι λίγοι στον αριθμό, αφήνοντας, έτσι, ελεύθερους και διαθέσιμους αρκετούς από τους πόρους του επεξεργαστή για οποιαδήποτε μελλοντική χρήση ή και επέκταση της αρχιτεκτονικής. Πιο συγκεκριμένα, οι πόροι του 'C31 που χρησιμοποιήθηκαν είναι οι ακόλουθοι:

- **8 registers** (4 extended-precision & 4 auxiliary). Ο λόγος που δεν χρησιμοποιήθηκε ένας συγκεκριμένος τύπος καταχωρητών ήταν ο λογικός διαχωρισμός των control και data operations.
- **2 32-bit timers** για τον υπολογισμό των τιμών των δύο βασικών σημάτων σύγκρισης (Timer0 για το ημίτονο, Timer1 για το τρίγωνο).
- **256 θέσεις εσωτερικής μνήμης** για την καταχώρηση των τιμών του βασικού ημιτόνου (ο αριθμός των θέσεων μπορεί να αλλάξει ανάλογα με τις απαιτήσεις, π.χ. μεγαλύτερη δειγματοληψία ημιτόνου, Three-Phase PWM, κ.ά).
- **To primary data bus** για τη λήψη των αποτελεσμάτων εξόδου.

Από τα παραπάνω προκύπτει ότι έχει καταναλωθεί το 50% των general-purpose registers, το 12,5% της data memory (RAM0-1: 2Kx32bits = 8Kbytes) και το σύνολο των timers του 'C31. Για οποιαδήποτε μελλοντική επέκταση, η επιπλέον επιβάρυνση σε πόρους αφορά αποκλειστικά και μόνο το Register File και τη μνήμη. Αυτό κάνει τον κώδικα πιο «ελαφρύ» και, φυσικά, περισσότερο ευέλικτο. Θα ακολουθήσει ανάλυση του τρόπου υλοποίησης της PWM με μονοπολική τάση εξόδου, με διπολική τάση εξόδου αλλά και του Three-Phase PWM.

2. Ανάλυση κώδικα

Ο αλγόριθμος υλοποίησης της PWM με Unipolar voltage switching φαίνεται στο παρακάτω flow chart.



Σχ.3.1 Flow Chart του αλγόριθμου υλοποίησης της μεθόδου PWM με μονοπολική τάση εξόδου

Ξεκινώντας την ανάπτυξη του κώδικα, γίνεται include το αρχείο “C3XMMRS.ASM”, το οποίο ορίζει τις θέσεις μνήμης, μεταξύ άλλων, και των τριών memory-mapped registers, που χρησιμοποιούν οι δύο timers του DSP. Κατόπιν, ορίζεται η αρχική διεύθυνση του κώδικα, ώστε να είναι γνωστή η θέση μνήμης από όπου θα αρχίσει να «διαβάσει» ο assembler.

```
.include "C3XMMRS.ASM"
```

```
.start "TINT_3",0x809802 ; Start assembling here
```

Προηγείται των βασικών ρουτίνων, ο καθορισμός των συχνότητων των δύο timers, που έχουν άμεση σχέση με τις συχνότητες των δύο σημάτων, του ημιτόνου και του τριγωνικού παλμού. Έτσι, ενεργοποιείται το 14^ο bit του Status Register (ST), το GIE (Global interrupt enable), θέτοντάς το 1. Με αυτό τον τρόπο, η CPU αντιδρά σε κάθε enabled interrupt.

```
GIE .set 0x00002000 ; This ST bit enables interrupts
```

Κατόπιν, προσδιορίζονται οι συχνότητες των δύο timers. Οι συχνότητές τους έχουν άμεση σχέση με τις συχνότητες των δύο σημάτων (τριγώνου και ημιτόνου) που υλοποιούν την PWM, οπότε ο σωστός και ακριβής καθορισμός τους είναι υψίστης σημασίας για την ορθή υλοποίηση της μεθόδου.

Εάν, για παράδειγμα, θέλουμε να εφαρμόσουμε ημίτονο συχνότητας 50 Hz και τριγωνική κυματομορφή συχνότητας 450 Hz, τότε είναι σαφές ότι αναφερόμαστε σε συντελεστή διαμόρφωσης συχνότητας $m_f = 9$. Έχουμε:

α) Για την τριγωνική κυματομορφή:

$$f_{triangle} = 450\text{Hz}, \text{ οπότε: } T_{triangle} = \frac{1}{f_{triangle}} = \frac{1}{450\text{Hz}} \approx 2.22\text{m sec}.$$

Στον κώδικα που περιγράφεται στο κεφάλαιο αυτό, ο Timer1, ο οποίος υλοποιεί το τρίγωνο, παράγεται με τη βοήθεια ενός counter. Αυτό που έχει σημασία για την εύρεση της συχνότητας του Timer1 είναι το γεγονός ότι η δειγματοληψία για μία περίοδο τριγώνου είναι 256 θέσεων, δηλαδή το interrupt του Timer1 «χτυπά» 256 φορές για μία περίοδο του τριγώνου. Όποτε συμβαίνει αυτό, εκτελείται η ρουτίνα του Timer1, όπου παράγονται οι τιμές του σήματος, κάτι που θα αναφερθεί διεξοδικά παρακάτω. Έτσι, η συχνότητα του τριγωνικού σήματος έχει την εξής σχέση με τη συχνότητα του interrupt:

$$T_{int,1} = \frac{T_{triangle}}{256} = \frac{2.22\text{m sec}}{256} = 8.671875\mu\text{sec}, \text{ άρα } f_{int,1} = \frac{1}{T_{int,1}} \approx 0.1153\text{MHz}$$

Στον κώδικα, «φορτώνεται» η τιμή 03C3h στη μεταβλητή TSTART.

```
TSTART .set 0x000003C3 ;reset and restart value for timers, CLOCK_MODE.
```

Η τιμή αυτή, στη συνέχεια, φορτώνεται σε κάθε ένα από τους Global-Control registers των δύο timers και καθορίζει την κατάσταση αυτών. Το 9^ο bit της μεταβλητής TSTART θέτει τους timers σε clock mode, ενώ το 10^ο bit (CLKSRC) καθορίζει το ρολόι των timers. Έχει την τιμή 1, οπότε ένα εσωτερικό ρολόι με συχνότητα ίση με το 1/2 της H1 συχνότητας

χρησιμοποιείται για να αυξήσει τον counter register του κάθε interrupt. Καθώς, λοιπόν, οι timers λειτουργούν σε clock mode, ισχύει:

$$\begin{aligned} f_{clock_mode} &= \frac{f_{timer_clock}}{2 * period_register} = \frac{f_{timer_clock}}{2 * PRD1} \Rightarrow PRD1 = \frac{f_{timer_clock}}{2 * f_{clock_mode}} = \\ &= \frac{f_{timer_clock}}{2 * f_{int,1}} = \frac{12.5MHz}{2 * 0.1153MHz} \approx 54.206 \Rightarrow 36h \end{aligned}$$

Γι' αυτό το λόγο, υπάρχει στον κώδικα η εντολή:

`PRD1 .set 36h ;f(triangle) = 450 Hz`

β) Για το ημίτονο:

$$f_{sine} = 50Hz, \text{ οπότε: } T_{sine} = \frac{1}{f_{sine}} = \frac{1}{50Hz} = 20msec.$$

Ο Timer0 σχετίζεται με την παραγωγή των τιμών του ημιτόνου. Ο τρόπος με τον οποίο αυτό γίνεται θα αναφερθεί παρακάτω. Προς το παρόν, σημασία έχει το γεγονός ότι η δειγματοληψία για την παραγωγή του ημιτόνου καθορίστηκε στις 256 τιμές, ώστε να υπάρξει μεγάλη ακρίβεια. Κάθε φορά που «χτυπάει» το interrupt του Timer0 (256 φορές για κάθε περίοδο του ημιτόνου), λαμβάνεται μία τιμή από τον LUT (Lookup Table) που περιέχει τις τιμές του σήματος. Η περίοδος του interrupt0 καθορίζεται από:

$$T_{int,0} = \frac{T_{sine}}{256} = \frac{20msec}{256} = 78.125\mu sec, \text{ άρα } f_{int,0} = \frac{1}{T_{int,0}} \approx 0.0128MHz$$

Αναφέρθηκε παραπάνω πως οι Timers λειτουργούν σε clock mode, επομένως ισχύει:

$$\begin{aligned} f_{clock_mode} &= \frac{f_{timer_clock}}{2 * period_register} = \frac{f_{timer_clock}}{2 * PRD0} \Rightarrow PRD0 = \frac{f_{timer_clock}}{2 * f_{clock_mode}} = \\ &= \frac{f_{timer_clock}}{2 * f_{int,0}} = \frac{12.5MHz}{2 * 0.0128MHz} \approx 488.28125 \Rightarrow 1E8h \end{aligned}$$

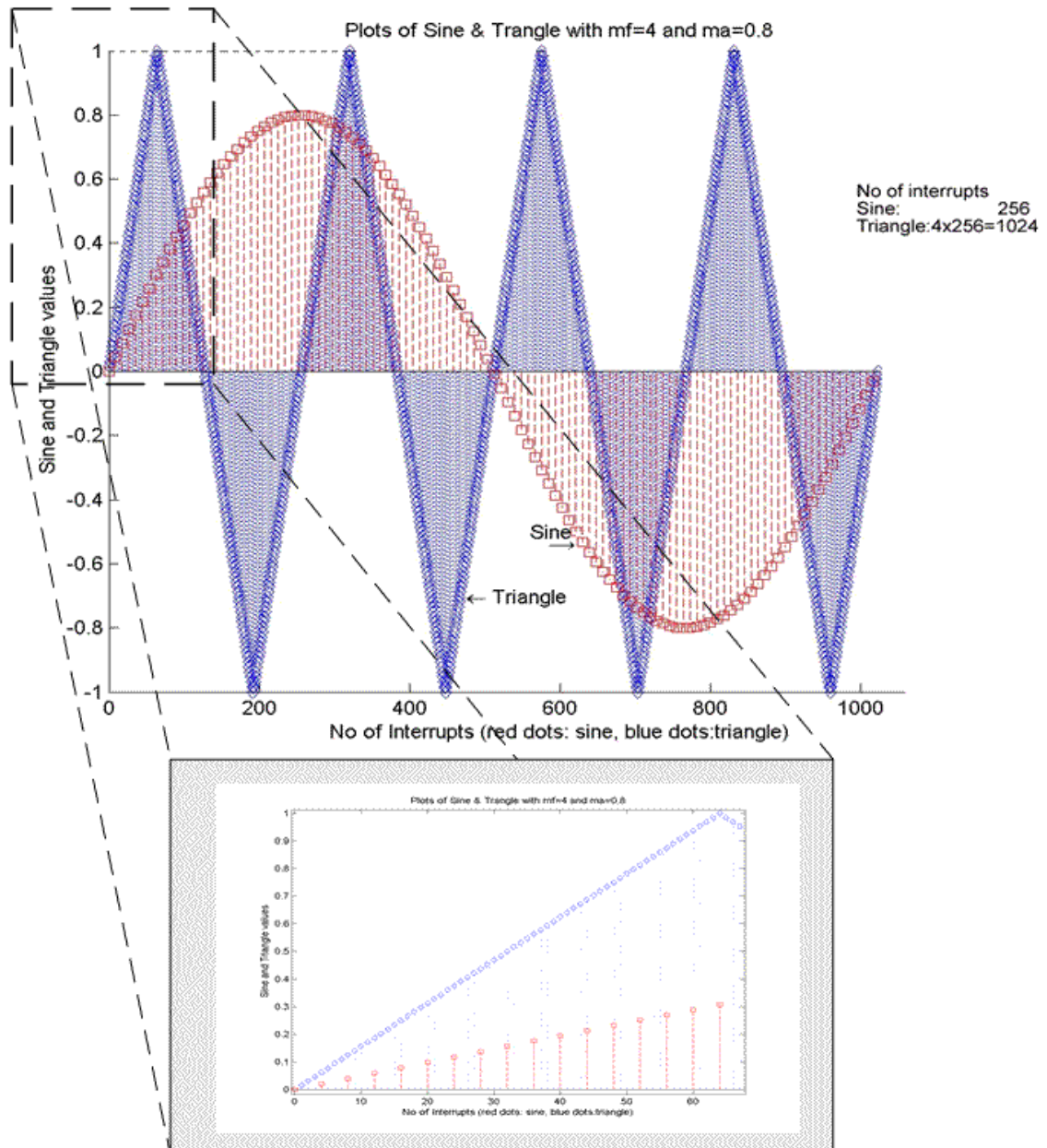
Έτσι, στον κώδικα υπάρχει η εντολή:

`PRD0 .set 1E8h ;f(sine) = 50 Hz`

Ο κώδικας περιλαμβάνει εναλλακτικές τιμές για τη συχνότητα των δύο σημάτων που διαχειρίζεται η PWM, έτσι ώστε να μπορούν να γίνουν οι απαραίτητες παρατηρήσεις για τη «συμπεριφορά» της μεθόδου. Ενδεικτικά αναφέρεται ότι ο κώδικας έχει πιστοποιηθεί για συχνότητα ημιτόνου ίση με 50 Hz, 60 Hz, αλλά και για τιμές μεγαλύτερες αυτών, ενώ, όσον αφορά στη συχνότητα της τριγωνικής κυματομορφής, ο κώδικας παρέχει την ευκολία της εναλλαγής της τιμής της από πολύ χαμηλά επίπεδα (π.χ. $f_{triangle} = 450Hz$) έως και υψηλότερα επίπεδα ($f_{triangle} = 25kHz$). Αναλυτική περιγραφή της αναγκαιότητας των παραπάνω δίνεται στο κεφάλαιο που περιλαμβάνει τα αποτελέσματα της πιστοποίησης του κώδικα.

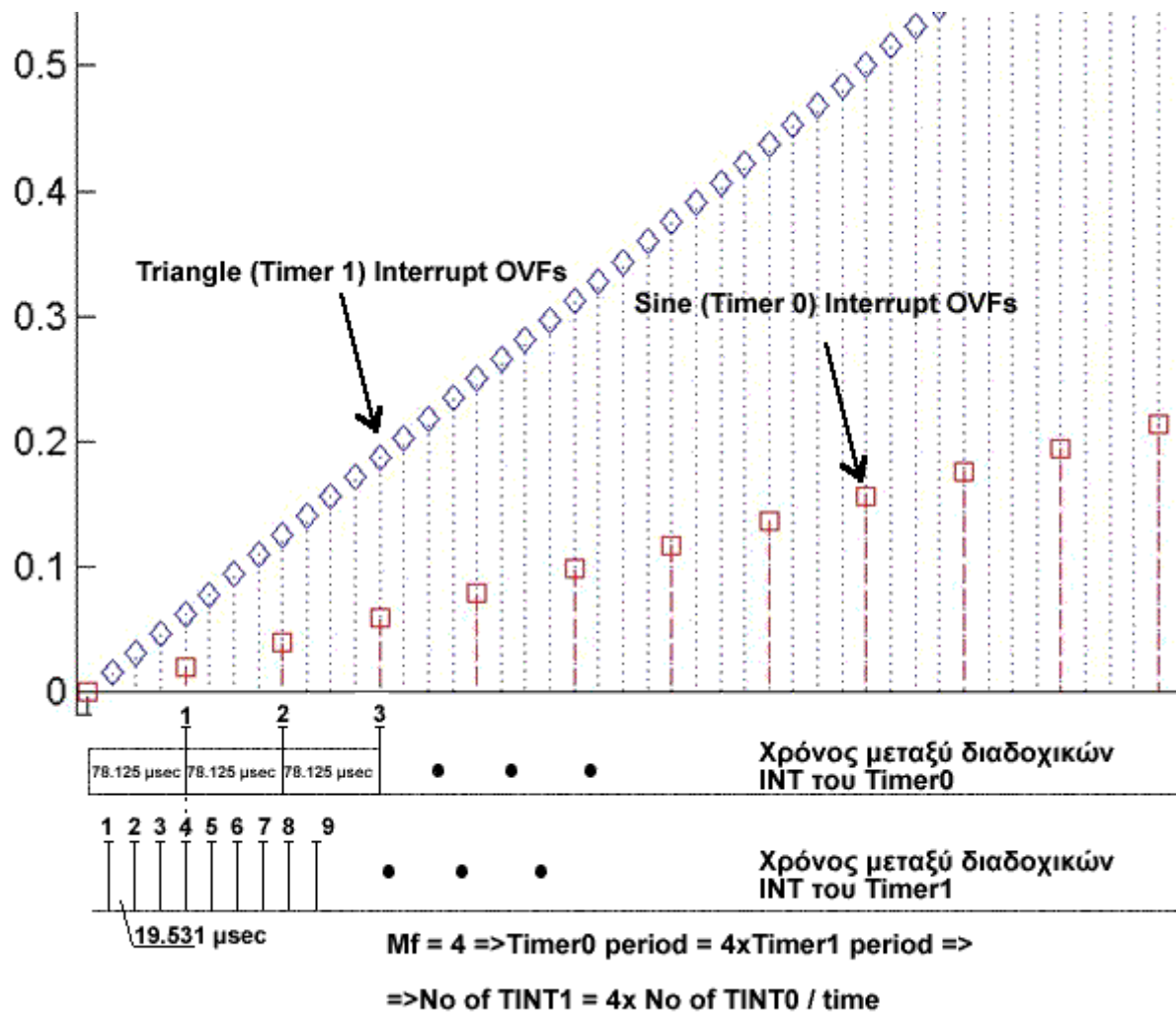
Ακολουθεί σχηματική αναπαράσταση του συσχετισμού των δύο Timers και πώς τα αντίστοιχα interrupts εμφανίζονται κατά τη διάρκεια εκτέλεσης του κώδικα:

Στο Σχ.3.2 παρατηρείται ο συνδυασμός των δύο κυματομορφών για μία πλήρη περίοδο του ημιτόνου με $m_f = 4$ και $m_a = 0.8$.



Σχ.3.2 Συσχετισμός των δύο βασικών σημάτων της μεθόδου PWM για μία περίοδο ημιτόνου

Το παρακάτω σχήμα δείχνει σε μεγέθυνση τον συσχετισμό που προαναφέρθηκε.



Σχ.3.3 Συσχετισμός των δύο Timers και χρόνοι μεταξύ διαδοχικών INTS

Αφού, λοιπόν, καθορίστηκαν οι περίοδοι των δύο σημάτων και διαμορφώθηκαν κατάλληλα οι Period registers των δύο timers, ακολουθεί η ανάλυση των ρουτίνων που υλοποιούν την PWM μέθοδο.

Όσον αφορά στην ημιτονοειδή κυματομορφή, αυτή παράγεται με τη χρήση της:

$$\text{float } 0.8 * \sin(\text{offs} * PI2N), \text{ όπου } PI2N = \frac{2 * PI}{N}$$

όπου, όπως φαίνεται, το πλάτος του ημιτόνου, και κατά συνέπεια, και ο συντελεστής m_a , μπορούν εύκολα να τροποποιηθούν ανάλογα με τις εκάστοτε συνθήκες. Αυτό γίνεται γιατί, κατά την παραγωγή του τριγώνου, η μέγιστη τιμή αυτού είναι η τιμή 1, ώστε, ουσιαστικά, το m_a , να «εξαρτάται» από τον συντελεστή που πολλαπλασιάζεται με τη συνάρτηση \sin . Οι τιμές που παράγονται, «φορτώνονται» σε έναν κυκλικό buffer κι έτσι, δημιουργείται ο LUT, από όπου η βασική ρουτίνα σύγκρισης των δύο σημάτων παίρνει τιμές.

Είναι σημαντικό να αναφερθεί ότι η μέθοδος του lookup – table είναι, πιθανόν, η πιο ευέλικτη και απλή μέθοδος για την παραγωγή ημιτονοειδών κυματομορφών. Η τεχνική αυτή, με λίγα λόγια, περιλαμβάνει το διάβασμα μιας σειράς αποθηκευμένων τιμών δεδομένων, που αναπαριστούν διακριτά δείγματα της κυματομορφής που πρέπει να παραχθεί. Αρκετά δείγματα (στην προκειμένη περίπτωση 256) παράγονται και αποθηκεύονται ώστε να αναπαριστούν με ακρίβεια μία ολοκληρωμένη περίοδο του σήματος. Έτσι, το σήμα παράγεται διαβάζοντας αδιάκοπα από τις θέσεις μνήμης που περιέχουν τα δεδομένα χρησιμοποιώντας έναν κυκλικό pointer. Στην παρούσα εφαρμογή χρησιμοποιήθηκε ένας κυκλικός buffer, σημαντικός και χρήσιμος πόρος του DSP. Πρέπει, μόνο να σημειωθεί πως, όταν ο κυκλικός buffer «διαβάσει» και τις 256 τιμές που περιέχει, πρέπει να δοθούν στον κώδικα εντολές ώστε να ξανα - αρχίσει να διαβάζει από την αρχική του διεύθυνση. Εάν δε γίνει αυτό, διαβάζει από τυχαίες διευθύνσεις, οπότε περιέχει τιμές ακατάλληλες για την υλοποίηση της μεθόδου.

Ένας τέτοιος πίνακας ημιτονοειδούς σήματος περιέχει διακριτές τιμές «ίσης απόστασης» για μια περίοδο. Όπως, ήδη έγινε φανερό και από τα παραπάνω, ένας LUT ημιτόνου N-σημείων, μπορεί να υπολογιστεί με τη χρήση της:

$$x(n) = \sin\left(\frac{2\pi n}{N}\right), \quad n = 0, 1, \dots, N-1.$$

Κάθε φορά που «χτυπάει» το interrupt του Timer0 εκτελείται η ρουτίνα του Sinusoidal Timer. Εκεί, φορτώνεται η «επόμενη» τιμή ημιτόνου από τον LUT σε κάποιο register.

Στη ρουτίνα του Triangle Timer, παράγονται οι τιμές του σήματος φορέα. Η μέγιστη τιμή του τριγώνου, όπως προαναφέρθηκε, είναι η τιμή 1. Κάθε φορά που μπαίνει στη ρουτίνα, ο counter του τριγώνου αυξάνεται ή μειώνεται, ανάλογα με την περίπτωση, κατά 0,015625 (1/64), όταν η δειγματοληψία του σήματος θέλουμε να είναι 256 δειγμάτων. Υπάρχουν διαθέσιμες οι κατάλληλες εντολές ώστε η δειγματοληψία να μπορεί να αλλάξει με ευκολία σε 32 δείγματα, κάτι που είναι απαραίτητο όταν η συχνότητα του φέροντος έχει υψηλή τιμή (π.χ. 25KHz). Ακόμη, έχει οριστεί κατάλληλο flag, ώστε εάν ο counter φτάσει στην τιμή 1 (ή την ξεπεράσει), να αρχίζει να μετράει προς τα κάτω, προς το -1 και το αντίστροφο.

Τέλος, στην Pulse routine, γίνεται η σύγκριση των τιμών των δύο σημάτων ώστε να προκύψουν οι επιθυμητοί παλμοί εξόδου. Σύμφωνα με τη θεωρητική ανάλυση της SPWM μεθόδου, η σύγκριση του πρώτου ημιτόνου με το τρίγωνο «εξάγει» τον πρώτο παλμό εξόδου στο 1^ο bit του data port του DSP (primary bus). Η εξαγωγή του δεύτερου παλμού γίνεται στο

2^ο bit του data port. Για να παραχθεί αυτό το αποτέλεσμα, για να γίνει, δηλαδή, η 2^η σύγκριση, αντιστρέφονται οι τιμές του ημιτόνου με τη χρήση της εντολής *NEGF*. Έτσι, ο LUT που περιέχει τις τιμές του ημιτόνου εξυπηρετεί και τις δύο συγκρίσεις χωρίς να χρειαστεί κάποια αλλαγή σε αυτόν.

Στην παραπάνω ρουτίνα σύγκρισης, ανάλογα με το αποτέλεσμα αυτής, θα μπορούσε, εναλλακτικά να φορτώνεται στον IOF τιμή ανάλογα με την έξοδο που θέλουμε να έχουμε στα XF0 και XF1 pins του DSP.

Αυτά τα pins μπορούν να δηλωθούν είτε ως είσοδοι ή εξοδοι. Μπορούν, επίσης, να χρησιμοποιηθούν για Read/Write. Στο reset, γράφεται μηδέν (0) στον IOF register. Στη συγκεκριμένη περίπτωση, κατά το Reset του IOF, θα φορτωθεί η τιμή 34 (σε hex) ή 110100 (σε binary) στον IOF, θα δηλωθούν, δηλαδή, τα XF0 και XF1 pins ως εξοδοι δεδομένων με αρχική τιμή 0.

LDI 34,IOF

Όταν, λοιπόν, η σύγκριση των δύο σημάτων έχει αποτέλεσμα ίσο με 0, σύμφωνα πάντα με την PWM, τότε θα φορτωθεί στον IOF, και συγκεκριμένα στο XF0 pin, η τιμή 0. Στην αντίθετη περίπτωση, το XF0 θα έχει την τιμή 1. Έτσι, μπορούμε να δούμε το αποτέλεσμα της μεθόδου (παλμοί εξόδου) ως έξοδο του DSP, εναλλακτικά, στα XF0 και XF1 pins αυτού.

3. Παρατηρήσεις

Ο κώδικας υλοποίησης της μεθόδου SPWM φορτώθηκε στον TMS320C31 και πιστοποιήθηκε η ορθότητά του χρησιμοποιώντας, αρχικά, τον debugger του διαθέσιμου DSK. Η έξοδος του συστήματος ήταν ορατή σε παλμογράφο και ήταν η επιθυμητή.

Κατά την παραπάνω διαδικασία βγήκαν σημαντικά συμπεράσματα, τα οποία θα παρουσιαστούν στην παρούσα ενότητα.

Καταρχάς, πρέπει να διευκρινιστεί πως τα TCLK0 και TCLK1 pins του DSP, όταν λειτουργούν ως outputs, όπως και στο συγκεκριμένο κώδικα, εξάγουν τους παλμούς λειτουργίας των Timer0 και Timer1, αντίστοιχα. Έτσι, στον παλμογράφο τα pins αυτά του DSP, τα 120 και 122, έδειχναν έξοδο τους παλμούς των δύο timers στις σωστές συχνότητες, δηλαδή στις συχνότητες στις οποίες είχαν καθοριστεί να λειτουργούν.

Η έξοδος του συστήματος στα pins του data port του DSP είναι η επιθυμητή έξοδος της SPWM. Παρατηρήθηκαν οι παρακάτω, πολύ σημαντικές «συμπεριφορές» του κώδικα.

Μεταβάλλοντας τη συχνότητα της τριγωνικής κυματομορφής και, κατά συνέπεια, τη συχνότητα λειτουργίας του Timer1, και με σταθερή τη συχνότητα του ημιτόνου στα 50 Hz, μετρήθηκε πειραματικά μέχρι ποια τιμή του συντελεστή συχνότητας m_f της μεθόδου υπάρχει έξοδος. Ξεκινώντας από πολύ μικρές συχνότητες για το τρίγωνο της τάξης μερικών εκατοντάδων Hz, η έξοδος παρατηρείται φυσιολογικά. Αυξάνοντας τον συντελεστή m_f και, κατ' επέκταση τη συχνότητα του φορέα, δεν παρατηρήθηκε κανένα πρόβλημα.

Στην περίπτωση που χρειαστεί η συχνότητα της τριγωνικής κυματομορφής να είναι της τάξης των 25KHz, ο κώδικας παρέχει τη δυνατότητα αυτό να συμβεί με ελάχιστες αλλαγές. Στην ουσία, αυτό που θα πρέπει να αλλάξει είναι η δειγματοληψία του τριγώνου. Εάν, αντί για 256 θέσεις δειγματοληψίας, εφαρμοστούν 64 θέσεις, τότε η έξοδος παρατηρείται φυσιολογικά στα 25KHz, δηλαδή με $m_f = 500$, πάντα με 50Hz συχνότητα ημιτονοειδούς σήματος.

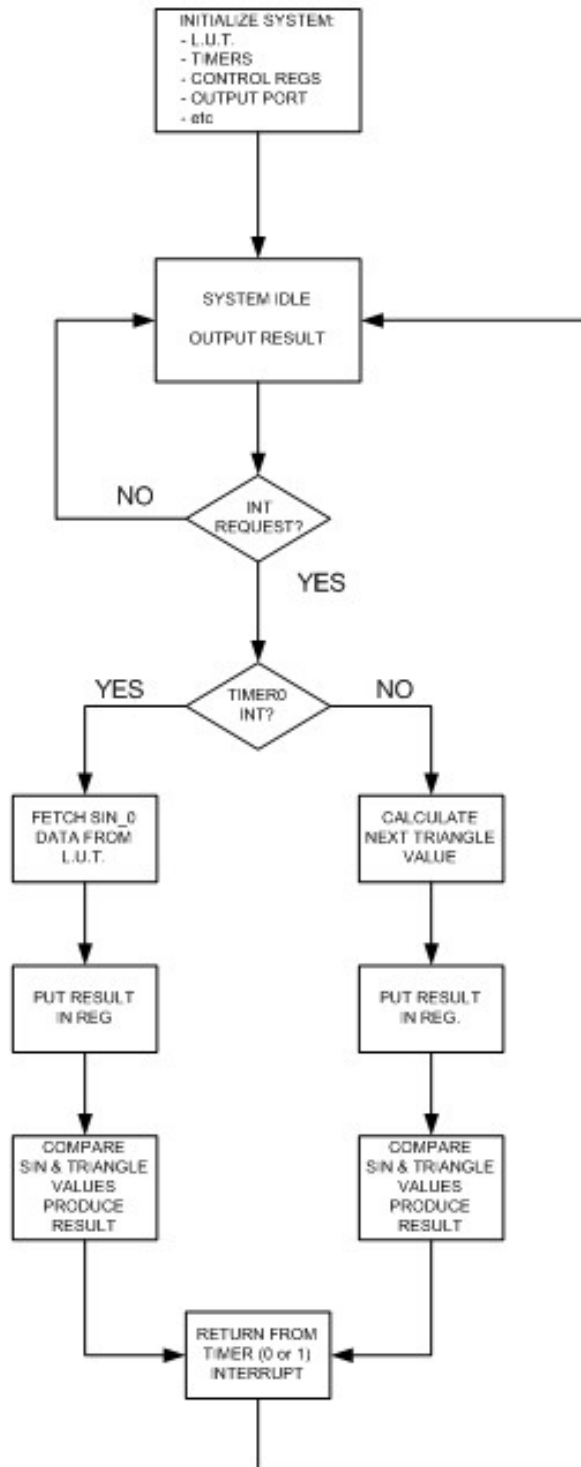
Τα παραπάνω οδηγούν στο εξής συμπέρασμα: εάν η συχνότητα του φορέα δεν χρειαστεί να είναι μεγαλύτερη των 1,6KHz, τότε ο κώδικας παρέχει το επιθυμητό αποτέλεσμα, με μεγάλη ακρίβεια, την ακρίβεια των 256 τιμών δειγματοληψίας σε μία περίοδο τριγώνου. Εάν, πάλι, η συχνότητα αυτή χρειαστεί να είναι πολύ μεγαλύτερη, τα τρίγωνα που θα «περιέχονται» σε μία περίοδο ημιτόνου είναι τόσα πολλά που, ακόμα και με 32 τιμές δειγματοληψίας για μια περίοδο τριγώνου, δεν θα υπάρξει σφάλμα ακρίβειας.

Από τα παραπάνω, προκύπτει ότι ο κώδικας που περιγράφεται για την υλοποίηση της μεθόδου SPWM στον TMS320C31, χρησιμοποιώντας τους πόρους του επεξεργαστή, μπορεί να εφαρμοστεί με τη σιγουριά του ορθού αποτελέσματος και της εύκολης τροποποίησής του ανάλογα με τις συνθήκες και τις ανάγκες της εφαρμογής.

4. Υλοποίηση της PWM με διπολική τάση εξόδου (PWM with Bipolar Voltage Switching).

Η διαφορά της μεθόδου με διπολική τάση εξόδου από την παραπάνω (με μονοπολική τάση εξόδου) έγκειται στο γεγονός ότι γίνεται σύγκριση ενός μόνο ημιτόνου με την τριγωνική κυματομορφή (Σχ.2.7).

Ο αλγόριθμος υλοποίησης της PWM με Bipolar voltage switching φαίνεται στο παρακάτω flow chart.



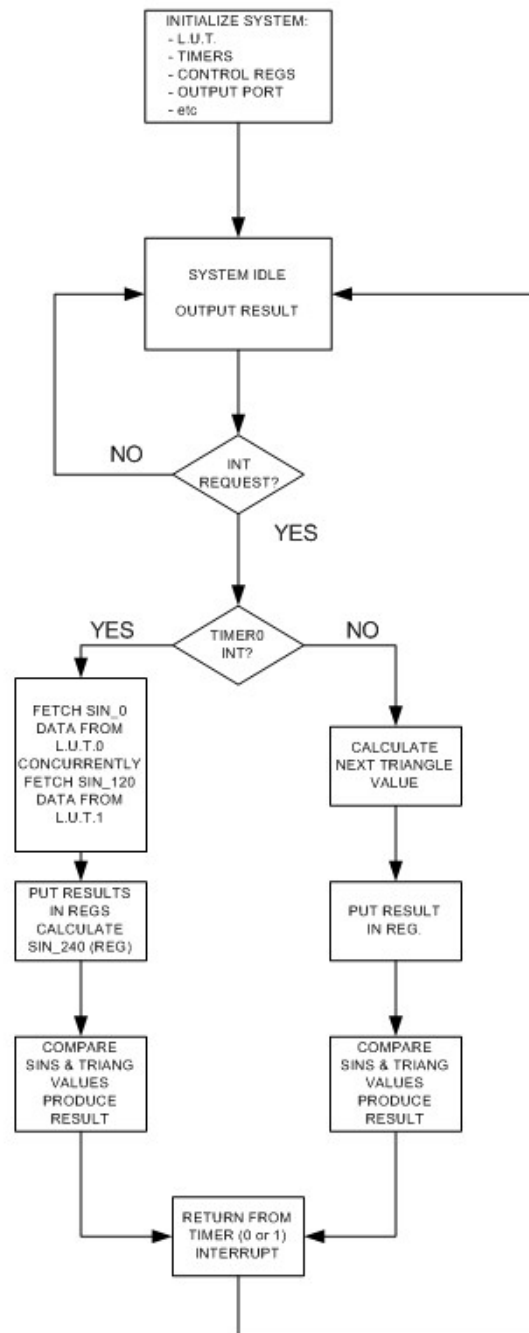
Σχ.3.4 Flow Chart του αλγόριθμου υλοποίησης της μεθόδου PWM με διπολική τάση εξόδου

Έτσι, στον κώδικα της μονοπολικής PWM αρκεί να εξαιρεθεί η σύγκριση του δεύτερου ημιτόνου με το τρίγωνο. Η έξοδος, πλέον, του 'C31 είναι μία, με τις τιμές που αντιστοιχούν στον 1 του Σχ.2.7 να «βγαίνουν» στο 1^ο pin του data port και αυτές που

αντιστοιχούν στο -1 του αντίστοιχου σχήματος να εμφανίζονται στο 1^ο pin του data port. Κατόπιν, γίνεται κατάλληλα επεξεργασία στις δύο παραπάνω εξόδους του DSP, κάτι που θα αναφερθεί στο κεφάλαιο των πειραματικών μετρήσεων.

5. Επέκταση του κώδικα για υλοποίηση three – phase SPWM.

Ο αλγόριθμος υλοποίησης της PWM με Bipolar voltage switching φαίνεται στο παρακάτω flow chart.



Σχ.3.5 Flow Chart του αλγόριθμου υλοποίησης της μεθόδου Three-Phase PWM

Για την περίπτωση της Three – Phase SPWM, οι αλλαγές που χρειάζεται να γίνουν στον κώδικα της Single – Phase SPWM είναι οι παρακάτω:

Το γεγονός ότι τόσο οι τιμές του ημιτόνου όσο και του τριγώνου λαμβάνονται από ένα look-up table, βοηθά πολύ, καθώς είναι απλό να δημιουργηθούν οι τρεις φάσεις του ημιτόνου. Αρκεί να διαβάζονται οι τιμές από τον LUT με «διαφορά φάσης», δηλαδή, για τη δεύτερη φάση του ημιτόνου, αρκεί να ξεκινήσει το «διάβασμα» των τιμών από την 85^η θέση του LUT. Για μεγαλύτερη ευκολία, μπορούν να αποθηκευτούν οι τιμές της δεύτερης φάσης σε ένα δεύτερο LUT. Αν οι δύο LUT αποθηκευτούν ο καθένας σε διαφορετικό RAM block, θα μπορεί να διαβάζονται παράλληλες οι τιμές από αυτούς, εκμεταλλευόμενοι, έτσι, τη δυνατότητα παράλληλων εντολών του DSP.

Η τρίτη φάση του ημιτόνου προκύπτει με τη χρήση της πρώτης και δεύτερης φάσης αυτού.

Δηλαδή, έχουμε:

$$\begin{aligned}\sin(x - 120^\circ) &= \sin x * \cos 120^\circ - \cos x * \sin 120^\circ = -\frac{\sin x}{2} - \cos x * \sin 120^\circ \\ &= -\frac{\sin x}{2} - \frac{1}{2} \sin(x + 120^\circ) + \frac{1}{2} \sin(x - 120^\circ) = -\frac{1}{2} [\sin x + \sin(x + 120^\circ)] \Rightarrow \\ \sin(x - 120^\circ) &= -\frac{1}{2} [\sin x + \sin(x + 120^\circ)]\end{aligned}$$

Από τα παραπάνω, προκύπτει ότι με χρήση πρώτης και δεύτερης φάσης της τριγωνικής κυματομορφής, μπορεί να παραχθεί το three-phase PWM.

Η υλοποίηση της μεθόδου με τη χρήση επεξεργαστή DSP αποτελεί ιδιαίτερα βοηθητικό στοιχείο, καθώς υπάρχει η δυνατότητα της παράλληλης εκτέλεσης εντολών. Έτσι, υπάρχει η δυνατότητα να φορτωθεί η πρώτη και η δεύτερη φάση του ημιτόνου σε διαφορετικά memory blocks, από όπου θα διαβάζονται οι τιμές της κάθε φάσης παράλληλα. Ο τρόπος αυτός υλοποίησης παράγει ένα κώδικα γρήγορο, αναδεικνύει τη χρησιμότητα του DSP και, κυρίως, δεν κάνει απαραίτητη τη χρήση περισσότερων πόρων του επεξεργαστή από όσους έχουν ήδη χρησιμοποιηθεί για την παραγωγή του Single – Phase SPWM.

ΚΕΦΑΛΑΙΟ 4^ο : ΠΡΟΣΟΜΟΙΩΣΗ ΚΥΚΛΩΜΑΤΟΣ ΙΣΧΥΟΣ - ΑΠΟΤΕΛΕΣΜΑΤΑ

1. Προσομοίωση Κυκλώματος Ισχύος

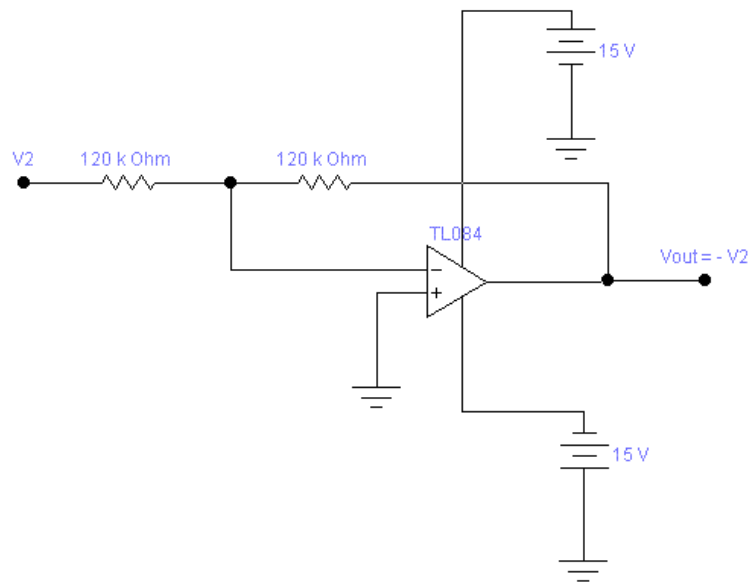
1.1 PWM με μονοπολική τάση εξόδου

Στο Κεφάλαιο 2 αναφέρθηκε ότι η έξοδος του αντιστροφέα κατά την PWM διαμόρφωση είναι επιθυμητό να είναι ημιτονοειδής και να υπάρχει η δυνατότητα ελέγχου του πλάτους και της συχνότητάς της. Το ημιτονοειδές σήμα ελέγχου έχει συχνότητα τη θεμελιώδη συχνότητα της τάσης εξόδου του αντιστροφέα, ενώ η συχνότητα της τριγωνικής κυματομορφής καθορίζει τη συχνότητα μετάβασης του αντιστροφέα.

Έως τώρα, εξετάστηκε η εξαγωγή των παλμών V_1 και V_2 , αποτέλεσμα της σύγκρισης του $V_{control}$ και του $-V_{control}$ αντίστοιχα, με την V_{tri} . Για να παραχθεί η ημιτονοειδής έξοδος της PWM μεθόδου, θα πρέπει να περάσει από ένα χαμηλοπερατό φίλτρο το σήμα: $V_1 - V_2$.

Το κύκλωμα, λοιπόν, που κατασκευάστηκε ώστε να παραχθεί το ημίτονο εξόδου της μεθόδου, εκτός από το χαμηλοπερατό φίλτρο, περιλαμβάνει και κύκλωμα που να έχει ως έξοδο την απαραίτητη είσοδο του φίλτρου, δηλαδή το σήμα $V_1 - V_2$.

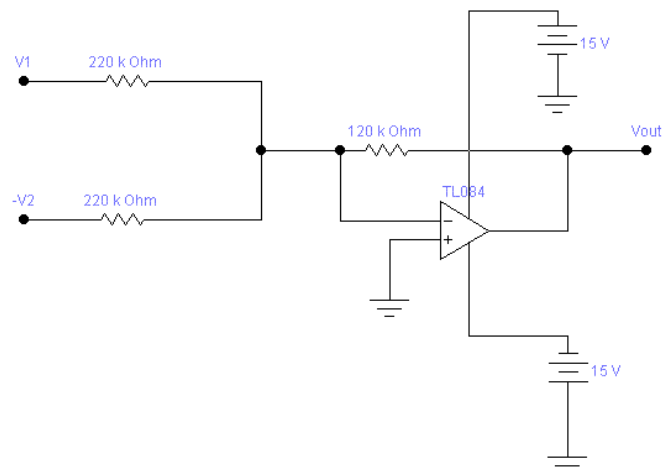
Αρχικά, παράχθηκε το $-V_2$ με τη χρήση ενός τελεστικού ενισχυτή να λειτουργεί ως αναστροφέας. Κατασκευάστηκε το κύκλωμα του Σχ.4.1.



Σχ.4.1 Κύκλωμα αναστροφής του σήματος V_2

Εφόσον οι αντιστάσεις σειράς του παραπάνω κυκλώματος είναι ίσες, ο αναστρέφων ενισχυτής έχει κέρδος τάσης ίσο με 1.

Στη συνέχεια, και αφού ήταν διαθέσιμο το σήμα $-V_2$, παράχθηκε το $V_1 - V_2$ με τη βοήθεια του αθροιστή δύο εισόδων του Σχ.4.2. Η μία είσοδος του αθροιστή είναι η V_1 και η άλλη είναι η έξοδος του αναστρέφοντα ενισχυτή. Η έξοδος είναι: $V_{out} = V_1 + (-V_2)$.



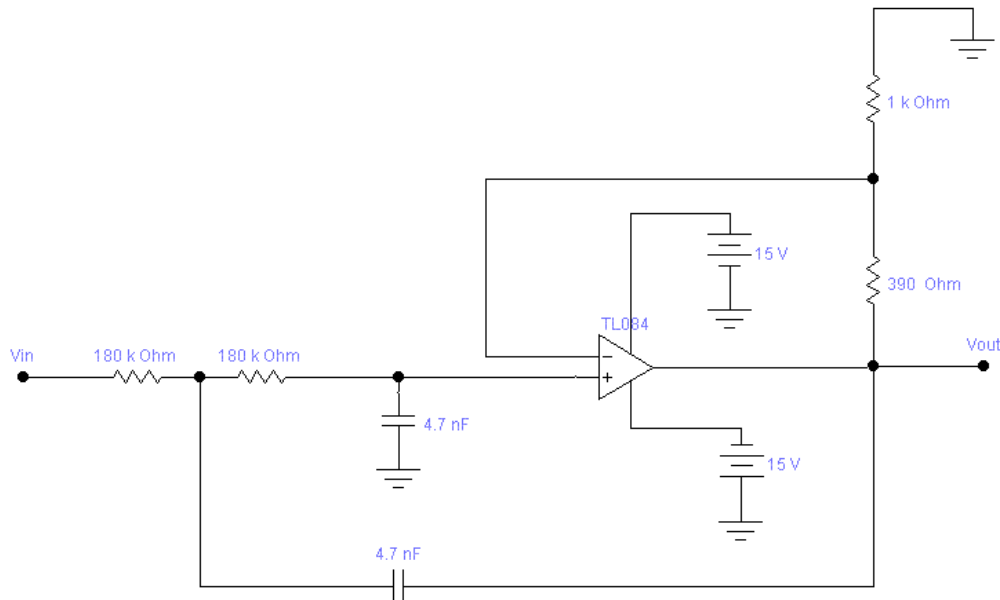
Σχ.4.2 Κύκλωμα αθροιστή

Ο αθροιστής στην έξοδό του, εάν είχε κέρδος ίσο με 1, δηλαδή εάν οι αντιστάσεις του κυκλώματος ήταν ίσες μεταξύ τους, θα έβγαζε σήμα εξόδου από +5V έως -5V. Με σκοπό το σήμα εξόδου του να έχει πλάτος ίσο με 5V (p-p), οι αντιστάσεις επιλέχθηκαν έτσι ώστε το κέρδος τάσης για το κάθε ένα από τα σήματα εισόδου να είναι ίσο με 0,5.

Τέλος, χρησιμοποιήθηκε ενεργό, χαμηλοπερατό φίλτρο Butterworth δύο πόλων ώστε να προκύψει το τελικό ημιτονοειδές σήμα. Ουσιαστικά, με τη βοήθεια τελεστικού ενισχυτή, έγινε δυνατή η χρήση ενεργού RC φίλτρου. Παρακάτω, δίνεται το κύκλωμά του.

Το ενεργό φίλτρο του παραπάνω σχήματος επιτρέπει τη διέλευση όλων των συχνοτήτων από μηδέν ως τη συχνότητα αποκοπής. Αυτή ισούται με:

$$f_c = \frac{1}{2\pi RC}.$$



Σχ.4.3 Κύκλωμα χαμηλοπερατού φίλτρου δύο πόλων

Το κέρδος τάσης κλειστού-βρόχου έχει οριακή τιμή ίση με $A = 1,586$, οπότε:

$$1.586 = \frac{R_1}{R_2} + 1,$$

$$\text{έτσι: } R_1 = 0.586R_2.$$

Στη συχνότητα αποκοπής, το συνολικό κέρδος τάσης μειώνεται κατά 3dB, δηλαδή ισχύει η:

$$V_{out} = 0.707 A V_{in}$$

Πάνω από τη συχνότητα αποκοπής, το κέρδος τάσης ελαττώνεται 40 dB ανά δεκάδα.

Στη συγκεκριμένη περίπτωση, θέλοντας η έξοδος ως προς την είσοδο να μην παρουσιάζει διαφορά πλάτους, καθορίστηκε ότι:

$$V_{out} = 0.707 A V_{in} \Rightarrow A = \frac{1}{0.707} \Rightarrow A \approx 1.41$$

Από τις παραπάνω σχέσεις προκύπτουν οι τιμές των στοιχείων του κυκλώματος του φίλτρου. Η συχνότητα αποκοπής του, για τιμή του συντελεστή $m_f = 9$, καθορίστηκε στα 188,22 Hz. Κι αυτό γιατί η πρώτη αρμονική του σήματος εισόδου του φίλτρου παρουσιάζεται στα 450Hz, αρμονική που δεν θα περάσει από το κύκλωμα του Σχ.4.3.

Στην περίπτωση του $m_f = 1$ και $f_{\sin e} = 50\text{Hz}$, η συχνότητα αποκοπής του φίλτρου καθορίστηκε στα 49,82Hz, ενώ για $m_f = 500$ και $f_{\sin e} = 50\text{Hz}$, η συχνότητα αποκοπής καθορίστηκε στα 1026Hz περίπου.

Για την υλοποίηση των πιο πάνω κυκλωμάτων χρησιμοποιήθηκε ο τελεστικός ενισχυτής TL084 της Texas, ο οποίος περιέχει τέσσερις τελεστικούς ενισχυτές.

Είναι προφανές ότι το κύκλωμα που περιγράφεται προσομοιώνει τη γέφυρα, καθώς με είσοδο τα σήματα «οδήγησης» των διακοπών του inverter (έξοδοι του κώδικα που αναπτύχθηκε στο TMS320C31), η έξοδος είναι ημίτονο ελεγχόμενου πλάτους και συχνότητας.

1.2 PWM με μονοπολική τάση εξόδου

Ισχύει ότι:

$$V_o(t) = V_{Ao}(t) - V_{Bo}(t) = 2V_{Ao}(t),$$

δηλαδή, η έξοδος του DSP, πολλαπλασιάζεται επί 2 πριν περάσει από το φίλτρο. Το φίλτρο παραμένει το ίδιο με αυτό του Σχ.4.3, που χρησιμοποιήθηκε παραπάνω. Οι θεωρήσεις γι' αυτό ως προς τη συχνότητα αποκοπής δεν διαφέρουν από τις ήδη προαναφερθείσες.

Μιας και οι έξοδοι του DSP είναι και εδώ τα 2 πρώτα pins του data port του επεξεργαστή, με τα περιεχόμενα του 2^ο pin να χρειάζονται αναστροφή, χρησιμοποιήθηκε η ίδια συνδεσμολογία για το κύκλωμα με αυτή του μονοπολικού PWM.

2. Πειραματικά αποτελέσματα – Μετρήσεις

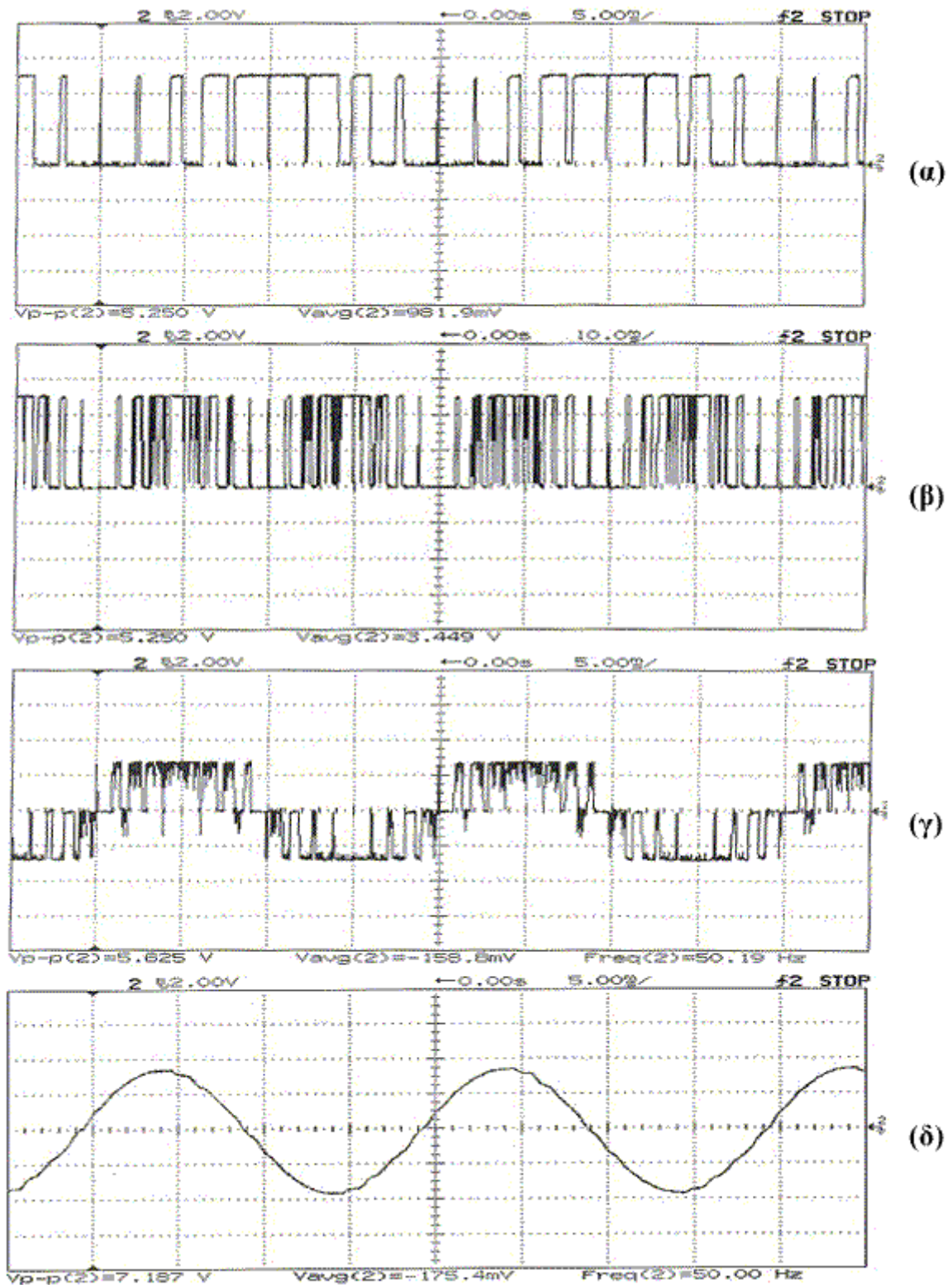
2.1 PWM με μονοπολική τάση εξόδου

Θα παρουσιαστούν τα αποτελέσματα του συστήματος που υλοποιήθηκε για τις ανάγκες της εφαρμογής.

Ως σύστημα αναφέρεται το Starter Kit του 'C31 και το κύκλωμα του φίλτρου. Ο επεξεργαστής έχει προγραμματιστεί με τον .asm κώδικα που περιγράφηκε στο 4^ο κεφάλαιο. Η έξοδος του kit αποτελεί είσοδο του κυκλώματος το οποίο, μεταξύ άλλων, περιέχει και το χαμηλοπερατό φίλτρο. Έξοδος, λοιπόν, του ολικού συστήματος αναμένεται να είναι ημίτονο ελεγχόμενου πλάτους και συχνότητας. Οι μετρήσεις – κυματομορφές που ακολουθούν έχουν καταγραφεί στον 54603B παλμογράφο της Helwett Packard για διάφορες περιπτώσεις τιμών της συχνότητας των δύο σημάτων εισόδου της PWM μεθόδου, αλλά και της συχνότητας αποκοπής του φίλτρου. Επιπλέον, έχει μετρηθεί και η παραμόρφωση του ημιτόνου εξόδου με τη βοήθεια του HM8027 Distortion Meter.

Παρατηρήθηκαν τα ακόλουθα:

$$\circ \quad f_{\sin e} = 50Hz, \quad f_{triangle} = 450Hz, \quad m_f = 9$$



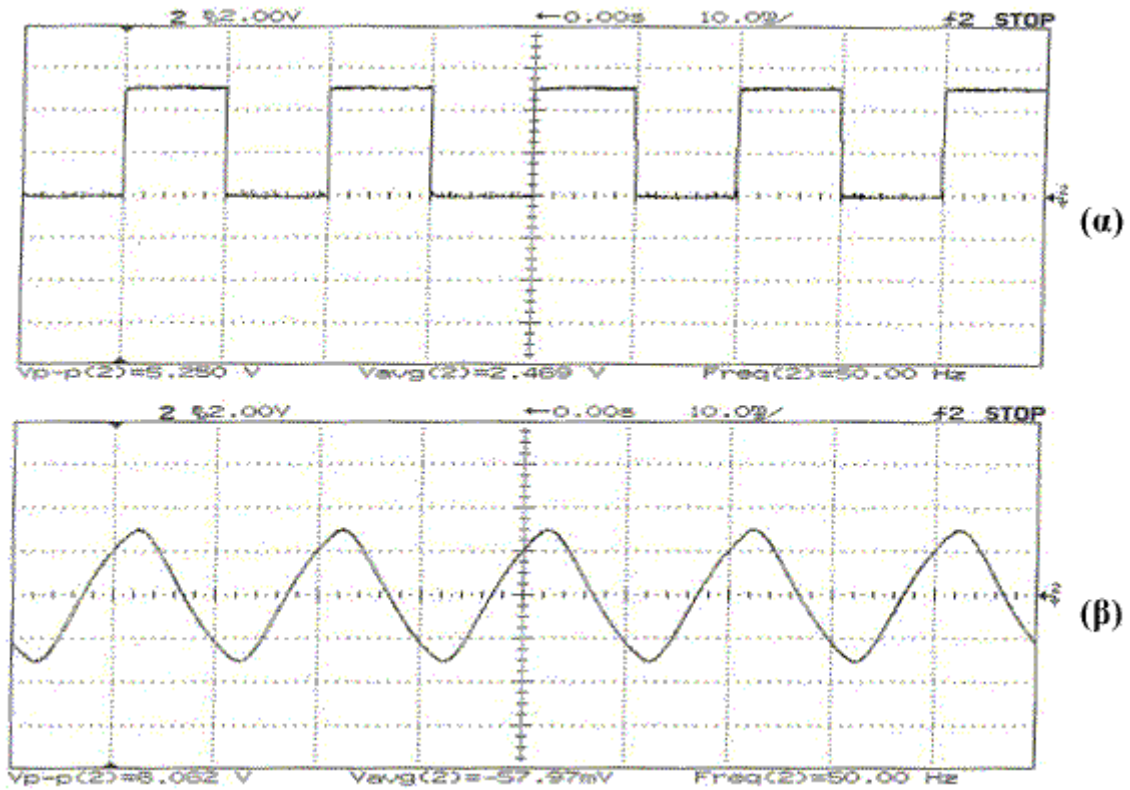
Σχ.4.4

α) Σήμα εξόδου V_1 του 'C31, β) Σήμα εξόδου V_2 του 'C31,

γ) Σήμα εισόδου του φίλτρου, δ) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)

Η παραμόρφωση του σήματος εξόδου του συστήματος μετρήθηκε 1,3%.

$$\circ \quad f_{\text{sin}} = 50\text{Hz} , f_{\text{triangle}} = 50\text{Hz} , m_f = 1$$



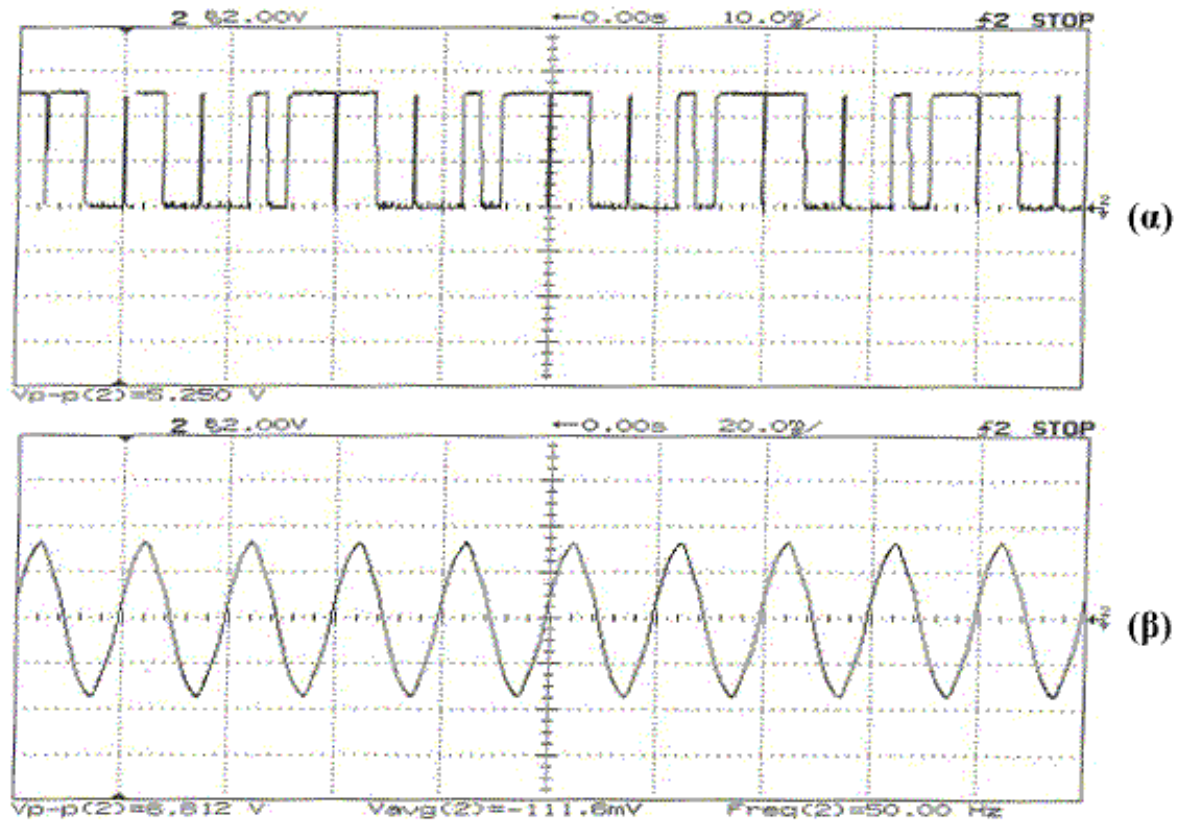
Σχ.4.5

α) Σήμα εξόδου V_1 του 'C31, β) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)

Όπως φαίνεται στο Σχ.4.5(α), η έξοδος του 1^{ου} pin του data port περιέχει ένα παλμό ανά περίοδο, πράγμα αναμενόμενο για $m_f = 1$.

Η παραμόρφωση του σήματος εξόδου του συστήματος μετρήθηκε 6%.

$$\circ \quad f_{\sin e} = 50\text{Hz} , f_{\text{triangle}} = 200\text{Hz} , m_f = 4$$



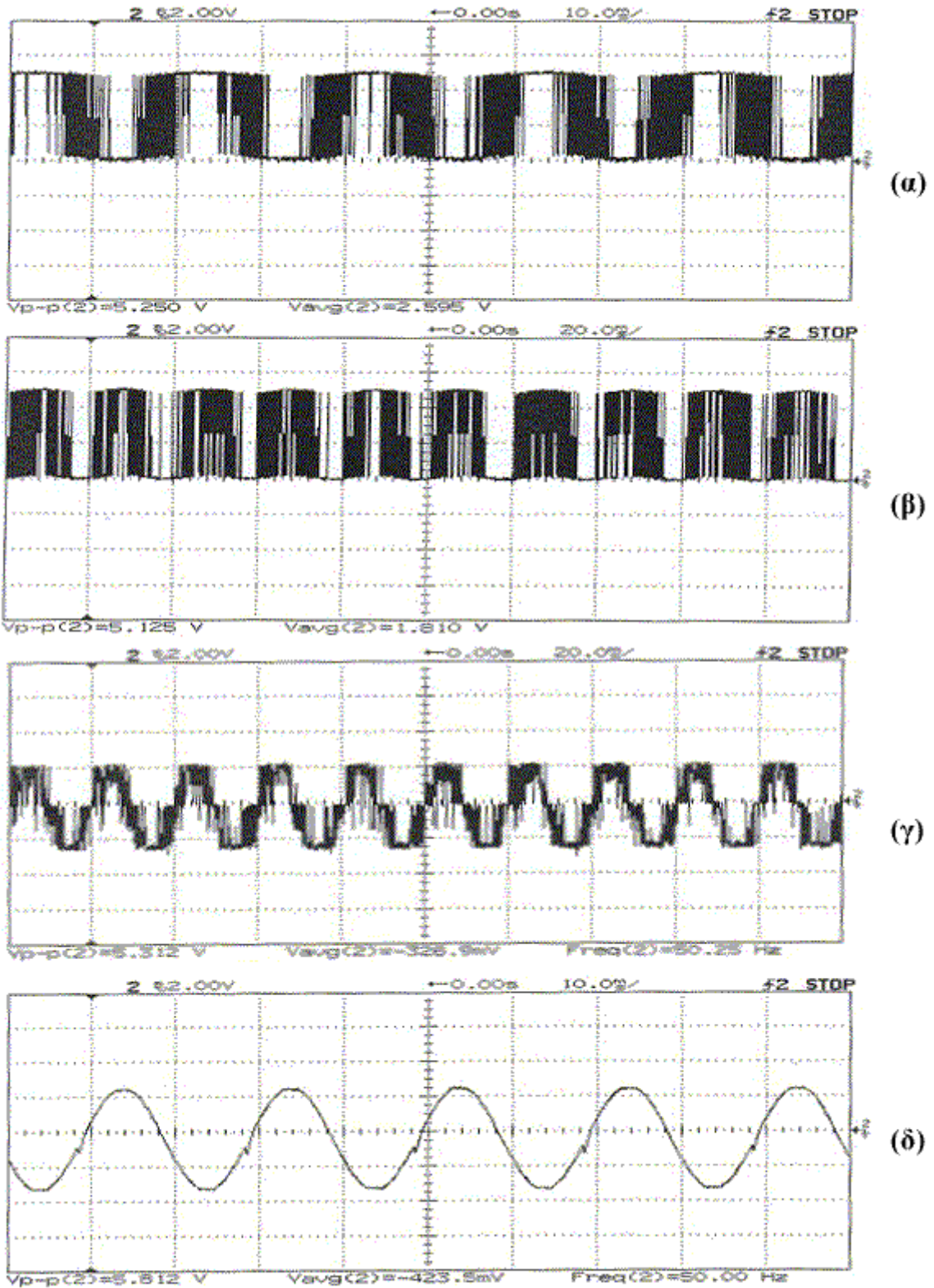
Σχ.4.6

α) Σήμα εξόδου V_1 του 'C31, β) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)

Στο Σχ.4.6(α) φαίνεται ότι η έξοδος του 1^{ου} pin του data port περιέχει τέσσερις παλμούς ανά περίοδο, πράγμα αναμενόμενο για $m_f = 4$.

Η παραμόρφωση του σήματος εξόδου του συστήματος μετρήθηκε 3.4%.

ο $f_{\sin e} = 50\text{Hz}$, $f_{\text{triangle}} = 25\text{KHz}$, $m_f = 500$

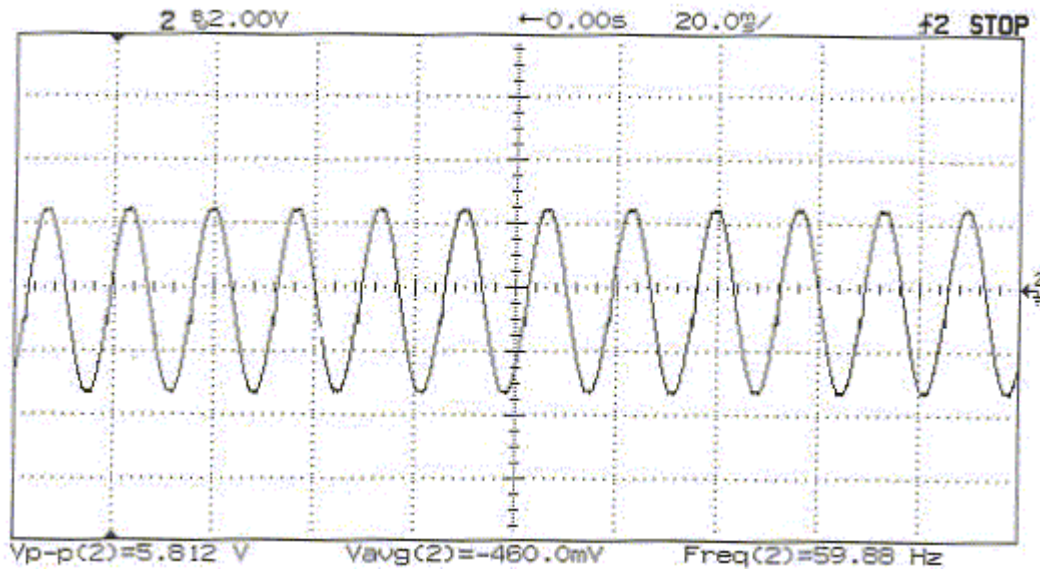


Σχ.4.7

- α) Σήμα εξόδου V_1 του 'C31, β) Σήμα εξόδου V_2 του 'C31,
 γ) Σήμα εισόδου του φίλτρου, δ) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)

Η παραμόρφωση του σήματος εξόδου του συστήματος μετρήθηκε 2%.

$$\circ \quad f_{\text{sine}} = 60\text{Hz}, \quad f_{\text{triangle}} = 25\text{KHz}, \quad m_f = 500$$



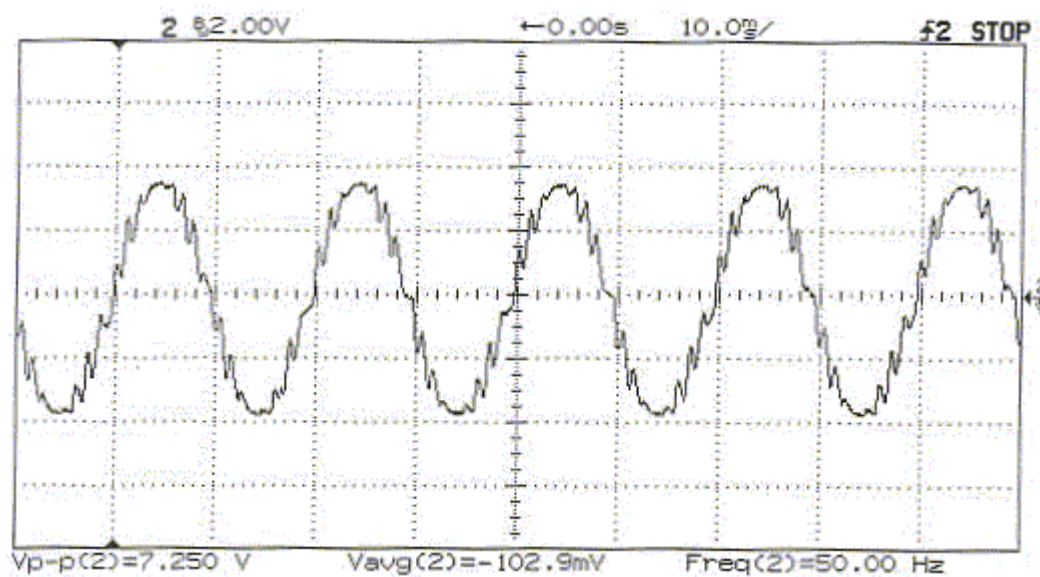
Σχ.4.8

α) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)

Η παραμόρφωση του σήματος εξόδου του συστήματος μετρήθηκε 1,9%.

Το χαμηλοπερατό φίλτρο δεν επιτρέπει την έλευση συχνοτήτων πάνω από τη συχνότητα αποκοπής. Οι παρακάτω μετρήσεις αποδεικνύουν πως όσο αυξάνεται η συχνότητα αποκοπής του φίλτρου πέραν της κατάλληλης συχνότητας για το αντίστοιχο m_f , τόσο και η έξοδος του φίλτρου εκφυλίζεται. Έτσι:

- Για $f_{\text{αποκοπής}} = 504\text{Hz}$, $f_{\text{sine}} = 50\text{Hz}$, $f_{\text{triangle}} = 450\text{Hz}$, $m_f = 9$, η έξοδος του συστήματος είναι η:



Σχ.4.9

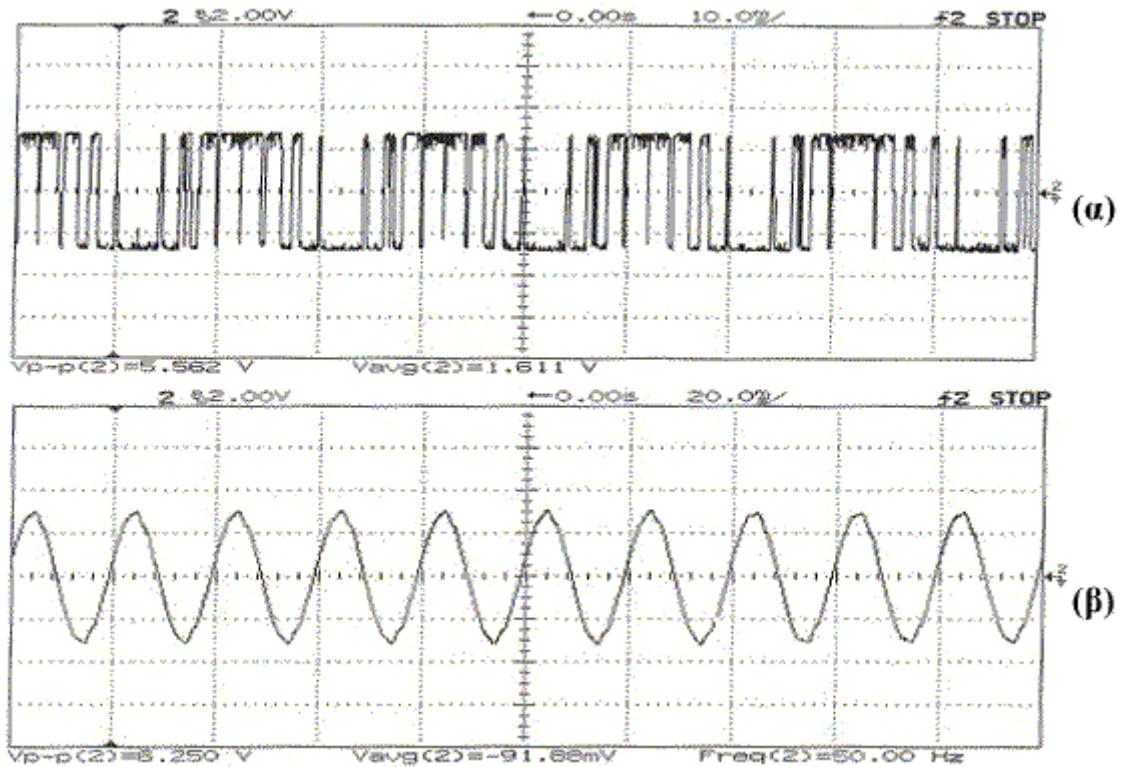
α) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)

Η παραμόρφωση του σήματος εξόδου του συστήματος μετρήθηκε 8%.

2.2 PWM με διπολική τάση εξόδου

Παρατηρήθηκαν τα ακόλουθα:

$$\circ f_{\sin e} = 50\text{Hz} , f_{\text{triangle}} = 450\text{Hz} , m_f = 9 , f_{\text{αποκοπής}} = 85\text{Hz}$$

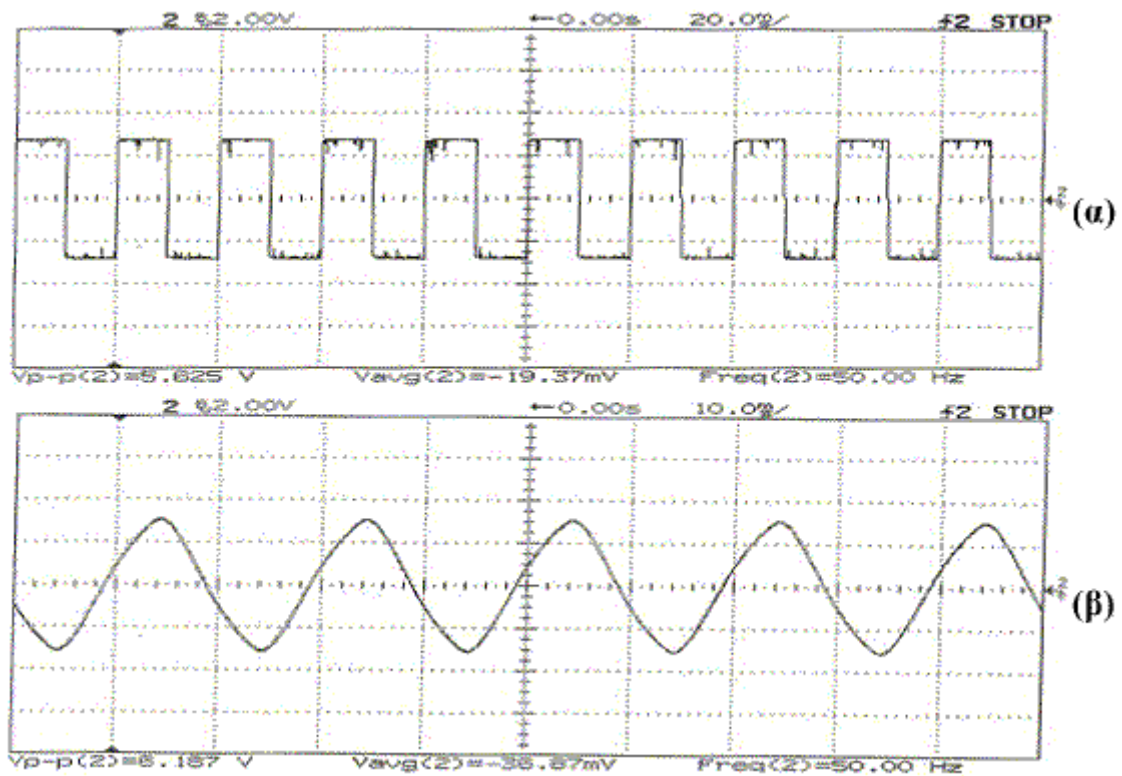


Σχ.4.10

α) Σήμα εξόδου V_1 του 'C31, β) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)

Η παραμόρφωση του σήματος εξόδου του συστήματος μετρήθηκε 2.5%.

$$\circ \quad f_{\sin e} = 50\text{Hz} , f_{\text{triangle}} = 50\text{Hz} , m_f = 1 , f_{\text{αποκοπής}} = 49,8\text{Hz}$$

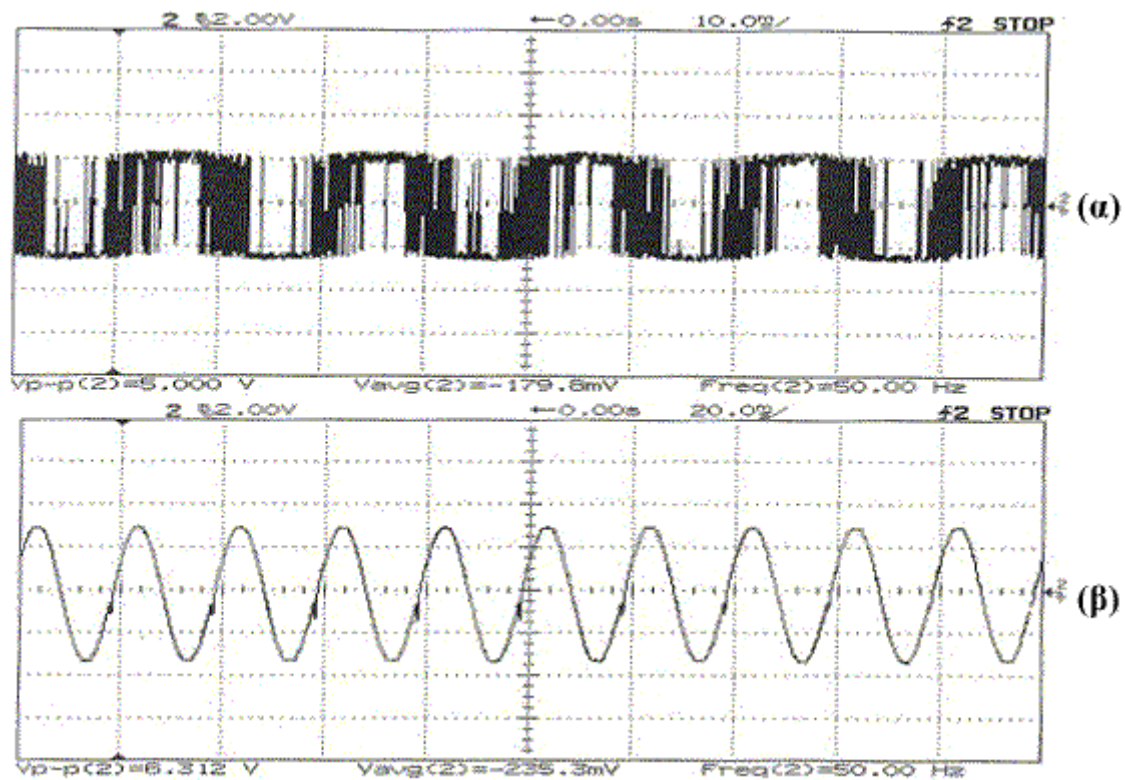


Σχ.4.11

α) Σήμα εξόδου V_1 του 'C31, β) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)

Η παραμόρφωση του σήματος εξόδου του συστήματος μετρήθηκε 5.6%.

$$\circ \quad f_{\sin e} = 50\text{Hz} , f_{\text{triangle}} = 25\text{KHz} , m_f = 500 , f_{\text{αποκοπής}} \approx 1\text{KHz}$$



Σχ.4.12

α) Σήμα εξόδου V_1 του 'C31, β) Ημίτονο εξόδου του συστήματος (έξοδος φίλτρου)

Η παραμόρφωση του σήματος εξόδου του συστήματος μετρήθηκε 2,2%.

ΚΕΦΑΛΑΙΟ 5^ο : *ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ*

1. Συμπεράσματα

Στην παρούσα διπλωματική εργασία υλοποιήθηκε επιτυχώς ένας PWM controller με χρήση του επεξεργαστή TMS320C31 της Texas Instruments. Πιο συγκεκριμένα, υλοποιήθηκαν οι μέθοδοι PWM with Unipolar Voltage Switching και PWM with Bipolar Voltage Switching, ενώ έγινε μελέτη του τρόπου υλοποίησης της Three-Phase PWWM.

Η έξοδος του συστήματος, ένα ημίτονο ελεγχόμενου πλάτους και συχνότητας, παρουσίασε παραμόρφωση που κυμαινόταν κατά μέσο όρο (λαμβάνοντας υπόψη όλες τις διαφορές παραμέτρους για τις οποίες πιστοποιήθηκε το σύστημα) στο 2%, τιμή καθ' όλα αποδεκτή.

Υλοποιήθηκε ένα σύστημα απόλυτο παραμετροποιημένο, μιας και για να αλλάξουν οι μεταβλητές της μεθόδου, ο χρήστης αρκεί να δώσει τιμές για τις περιόδους των δύο βασικών σημάτων, καθώς και του συντελεστή πλάτους.

2. Μελλοντικές επεκτάσεις

Τα παραπάνω μπορεί να γίνουν πιο εύκολα για το χρήστη, ως μελλοντική επέκταση, με την υλοποίηση μίας διεπικοινωνίας (interface) μεταξύ συστήματος και Η/Υ. Μέσω ενός γραφικού περιβάλλοντος (GUI, Graphical User Interface) ο χρήστης θα μπορεί να δώσει διαφορετικές παραμέτρους μέσω της σειριακής θύρας του DSP TMS320C31 για το PWM δυναμικά, χωρίς να χρειάζεται να γνωρίζει λεπτομέρειες της υλοποίησης.

Επιπλέον επέκταση θα ήταν αυτή η διεπικοινωνία να γίνεται εξ' αποστάσεως (μέσω δικτύου internet ή μέσω ασύρματης επικοινωνίας) καθώς ως επί το πλείστον οι Α/Γ και, κατ' επέκταση ο PWM controller, βρίσκονται σε απομακρυσμένα σημεία.

Το γραφικό περιβάλλον θα μπορούσε, επίσης, να ελέγχει τη σωστή λειτουργία του DSP λαμβάνοντας ανά τακτά χρονικά διαστήματα μέσω σειριακής feedback από το δεύτερο με στοιχεία της παρούσας κατάστασής του.

Ως μελλοντική επέκταση μπορεί να θεωρηθεί και η ανάπτυξη και υλοποίηση ενός Three-Phase PWM, το οποίο μάλιστα και αναπτύχθηκε ως κώδικας, απλώς δεν πιστοποιήθηκε το σύστημα πειραματικά, καθώς έπρεπε να αναπτυχθεί μία τελείως διαφορετική ηλεκτρονική διάταξη πιστοποίησής του, πράγμα που ξεφεύγει από τα πλαίσια της παρούσας διπλωματικής εργασίας.

Όλα τα παραπάνω μπορούν να υλοποιηθούν με χρήση του ίδιου επεξεργαστή, του TMS320C31.

Με αλλαγή του επεξεργαστή με κάποιο νεότερο μοντέλο με περισσότερες δυνατότητες, της ίδιας εταιρείας ή άλλης, οι μελλοντικές επεκτάσεις θα είναι περισσότερες, αφού υπάρχουν DSP που έχουν ενσωματωμένη PWM Unit (π.χ. Motorola MC68HC708), απελευθερώνοντας, έτσι, πόρους του επεξεργαστή (timers) που χρησιμοποιούνται με την παρούσα μέθοδο για την αλγοριθμική υλοποίηση του PWM.

ΠΑΡΑΡΤΗΜΑ Α: *TMS320C3x DSP Starter Kit*

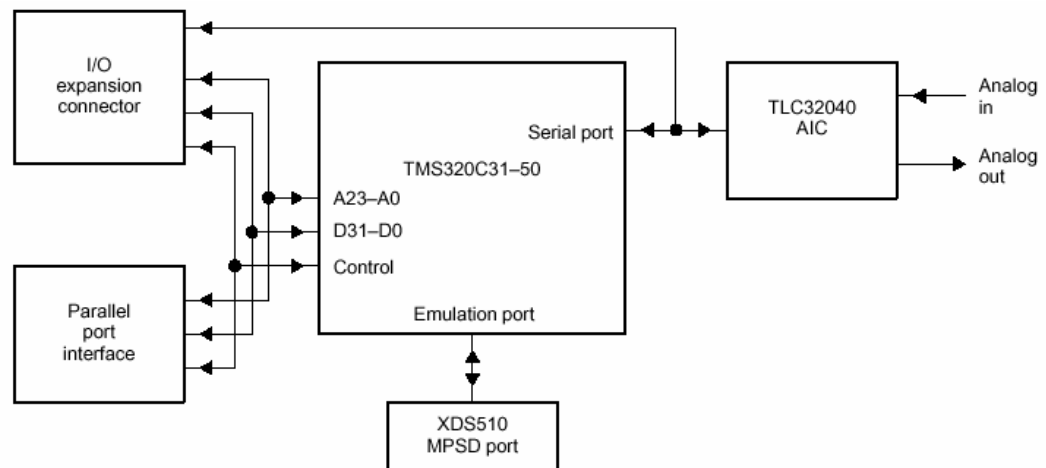
1. Εισαγωγή

Στο Παράρτημα θα γίνει αναφορά στο Starter Kit του TMS320C3x DSP (DSK). Το C3x DSK είναι ένα απλό αναπτυξιακό, χαμηλού κόστους και υψηλής απόδοσης, που μας επιτρέπει να πειραματιστούμε και να χρησιμοποιήσουμε τα TMS320C3x DSPs σε εφαρμογές πραγματικού χρόνου. Περιέχει ένα TMS320C31 για την επικύρωση του κώδικα. Το DSK, μας παρέχει τη δυνατότητα να δημιουργήσουμε τον κώδικα σε PC, να τον φορτώσουμε και να τον τρέξουμε στο αναπτυξιακό. Η επικύρωση – επιβεβαίωση του κώδικα γίνεται με τη βοήθεια του παρεχόμενου debugger, που βοηθά στην ανακάλυψη τυχόν λαθών σε αυτόν.

Το αναπτυξιακό του TMS320C3x DSP περιλαμβάνει :

- Ένα TMS320C31 DSP κινητής υποδιαστολής
- Παράλληλη θύρα (LPT), με την οποία συνδέεται με PC και επιτρέπει, έτσι, την επικοινωνία του TMS320C31 με προγράμματα του PC.
- 40 ns κύκλο εντολής, 50 MFLOPS, 25 MIPS
- Απόκτηση αναλογικών δεδομένων μέσω του TLC32040 (Analog Interface Circuit – AIC), καθώς αυτό περιέχει :
 - ο 14-bit δυναμικού εύρους ADC και DAC
 - ο μεταβλητό εύρος δειγματοληψίας των converters, που φτάνει και τα 20,000 δείγματα το δευτερόλεπτο
 - ο φίλτρο εισόδου για αντιμετώπιση φαινομένων aliasing και φίλτρο εξόδου για επανακατασκευή σήματος.
- Δύο RCA connectors για αναλογική είσοδο και έξοδο
- Connector για τον XDS510 emulator
- Συνδέσεις-θύρες επέκτασης των pins του TMS320C31 για χρήση του αναπτυξιακού σε συνδυασμό με άλλες κάρτες

Παρακάτω, θα δούμε συνοπτικά, ένα block diagram του TMS320C3x DSK hardware, όπως αυτό φαίνεται στο Σχ.Α.1.



Σχ.Α.1 TMS320C3x DSK Block Diagram

Τα βασικά μέρη του αναπτυξιακού είναι το TMS320C31 DSP, το TLC32040 AIC, οι θύρες επέκτασης, το ρολόι του συστήματος, η παράλληλη θύρα (printer port) και ένα LED τριών χρωμάτων.

Όπως φαίνεται και στο Σχ.Α.1, όλα τα σήματα του 'C3x οδηγούνται στις θύρες επέκτασης. Αυτές περιλαμβάνουν 4 32-pin headers, ένα 11-pin jumper και ένα 10-pin XDS510 header.

Το TLC32040 επικοινωνεί με το TMS320C31 μέσω μιας σειριακής θύρας. Ένα jumper επιτρέπει την κατάργηση της μεταξύ τους σύνδεσης και την επικοινωνία του αναπτυξιακού με άλλη, εξωτερική κάρτα.

Τέλος, οι δύο RCA connectors παρέχουν την αναλογική είσοδο και έξοδο, που τυχόν θέλουμε να έχουμε στην κάρτα του αναπτυξιακού.

2. Εγκατάσταση του DSK Assembler και Debugger

Η ενότητα αυτή περιγράφει τον τρόπο με τον οποίο γίνεται η εγκατάσταση του DSK σε ένα PC.

Προϋποθέσεις Υλικού (Hardware)

Το PC που θα φιλοξενεί το DSK θα πρέπει, εκτός της παράλληλης θύρας για την επικοινωνία PC-DSK και του κατάλληλου καλώδιου για την επίτευξη αυτής, να έχει το λιγότερο 64MB μνήμη. Προτίνεται έγχρωμη οθόνη, χωρίς αυτό να είναι απαραίτητο. Το αναπτυξιακό τροφοδοτείται μέσω μετασχηματιστή, που παρέχει 7-12 Vdc και τουλάχιστον 400-1500mA.

Προϋποθέσεις Λογισμικού (Software)

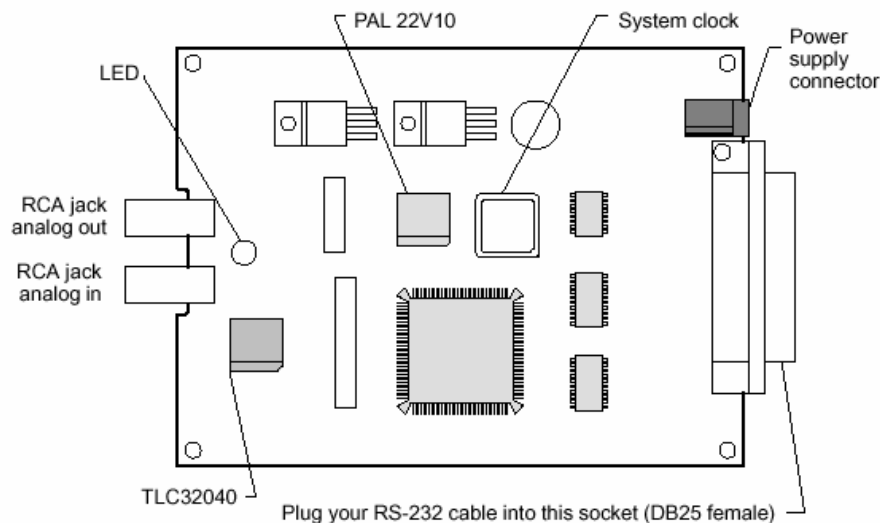
Απαιτείται MS-DOS και Windows 95/98/Me. Για την εγκατάσταση του DSK assembler, χρειάζεται το dsk3a.exe, ενώ, για να μπορέσουμε να τρέξουμε τον debugger του συστήματος, χρειαζόμαστε το αρχείο dsk3d.exe. Στο όλο πακέτο που περιέχει και το DSK, περιέχονται επιπλέον αρχεία με δείγματα πηγαίου κώδικα, καθώς και αρχεία με χρήσιμες πληροφορίες για τα αρχεία κώδικα που δίδονται, αλλά και περαιτέρω πληροφορίες για το DSK.

Παρακάτω, θα περιγράψουμε αναλυτικά τον τρόπο με τον οποίο γίνεται η εγκατάσταση του DSK στο PC.

Βήμα 1^ο : Σύνδεση του DSK με το PC

Όλα τα παρακάτω, φαίνονται με λεπτομέρειες στο Σχ.Α.2.

1. Έχουμε κλειστό τον Η/Υ.
2. Συνδέουμε το καλώδιο της παράλληλης με την παράλληλη θύρα (LPT) του υπολογιστή.
3. Συνδέουμε την άλλη άκρη του καλωδίου με την DSK DB25 θύρα.



Σχ.Α.2 Σύνδεση του DSK με το PC και την παροχή ρεύματος

4. Συνδέουμε τον μετασχηματιστή που περιγράψαμε παραπάνω, με το αναπτυξιακό και συνδέουμε το μετασχηματιστή με την τροφοδοσία.
5. Ανοίγουμε το PC.
6. Το LED του DSK θα πρέπει να εναλλάσσεται μεταξύ κόκκινου και πράσινου χρώματος.

Βήμα 2^ο : Εγκατάσταση του Λογισμικού του DSK

1. Δημιουργούμε στον σκληρό δίσκο ένα directory με το όνομα *dskttools*. Το directory αυτό θα περιέχει το software του DSK assembler και debugger.

2. Αντιγράφουμε τα περιεχόμενα της δισκέτας που ήταν μαζί με το DSK στο directory που μόλις φτιάξαμε.

Παρατήρηση: Καλό είναι να δημιουργήσουμε και ένα αντίγραφο της δισκέτας για κάθε ενδεχόμενο.

Βήμα 3^ο : Τροποποίηση του CONFIG.SYS (Optional)

Όταν χρησιμοποιούμε τον debugger, μπορούμε να έχουμε μόνο 20 αρχεία ανοιχτά ή ενεργά ταυτόχρονα. Για να πούμε στο σύστημα να μην επιτρέπει περισσότερα από 20 ενεργά αρχεία, πρέπει να προσθέσουμε την ακόλουθη γραμμή στο αρχείο config.sys:

FILES=20

Μόλις αλλάξουμε το Config.sys, πρέπει να επανα-εκκινήσουμε τον Η/Υ.

Βήμα 4^ο : Επαλήθευση της σωστής εγκατάστασης

Για να σιγουρευτούμε ότι έχουμε εγκαταστήσει σωστά το αναπτυξιακό, μαζί με τον assembler και τον debugger, εισάγουμε την ακόλουθη εντολή στο prompt του συστήματος:

dsk3d <ENTER>

Αφού εισάγουμε την παραπάνω εντολή, θα πρέπει να δούμε στην οθόνη μας το παράθυρο του debugger, όπως αυτό φαίνεται στο Σχ.Α.3.

DISASSEMBLY				C31 DSP STARTERS KIT			
809c03	50700080	start	LDIU 00080h,DP	PC	00809c03	SP	008098de
809c04	08349c2c		LDI @09c2cH, SP	R0	00000000	R1	00000000
809c05	07608000		LDF 0.000000e+00,R0	R2	00000000	R3	00000000
809c06	c610c1c0		LDI *AR0,R0 LDI *AR	R4	00000000	R5	00000000
809c07	c610c1c0		LDI *AR0,R0 LDI *AR	R6	00000000	R7	00000000
809c08	08600100		LDI 256,R0	AR0	00000000	AR1	00000000
809c09	09a09c00		LSH @09c00H,R0	AR2	00000000	AR3	00000000
809c0a	61809c0e		BRD jump	AR4	00000000	AR5	00000000
809c0b	07618000		LDF 0.000000e+00,R1	AR6	00000000	AR7	00000000
809c0c	07628000		LDF 0.000000e+00,R2	IR0	00000000	IR1	00000000
809c0d	07630000		LDF 1.000000e+00,R3	ST	00000000	RC	00000000
809c0e	07640000	jump	LDF 1.000000e+00,R4	RS	00000000	RE	00000000
809c0f	087b0003	loop	LDI 3,RC	DP	00000000	BK	00000000
809c10	64809c1a		RPTB block	IE	00000000	IF	00000000
809c11	02640001		ADDI 1,R4				
COMMAND				MEMORY			
Texas Instruments 1994				809800	00000007	fffffffc	00809802 00809827
				809804	0080982c	00809839	0080983c 0080983f
				809808	00809843	00809842	00809868 0080989a
				80980c	008098a9	10800000	0f350000 0f300000
				809810	0f200000	0f320000	0f280000 0f290000
				809814	1a770004	6a050006	628098a9 50700080
load testa							

Σχ.Α.3 Βασική οθόνη debugger

Όταν, πλέον, η εγκατάσταση έχει γίνει σωστά, που σημαίνει ότι η επικοινωνία του DSK και του PC είναι ορθή, το LED του αναπτυξιακού αναβοσβήνει με την εξής ακολουθία χρωμάτων : κόκκινο – κίτρινο – πράσινο – κόκκινο, ... Αυτό συμβαίνει γιατί οι timers του DSP αρχικοποιούνται.

Στην περίπτωση που δούμε στην οθόνη μας την εικόνα του Σχ.Α.4, τότε η εγκατάσταση του όλου συστήματος δεν έχει γίνει σωστά. Αυτό μπορεί να οφείλεται σε κακή εγκατάσταση του λογισμικού.

```
TESTING TMS320C3x DSK RESET AT PORT 0x378 (LPT1)
>>>> HPACK (ERROR pin) did not go high during reset
SELECT: 1) LPT1 0x378      (alternate LPT2)
        2) LPT2 0x278      (alternate LPT3)
        3) LPT3 0x3BC      (alternate LPT1)
        H) Additional online help

CHECK: TARGET POWER (LED IS RED OR GREEN)
      PORT SELECTION
      I/O CONNECTIONS AND CABLES
      POWER CONSERVATION SOFTWARE (LAPTOPS!)
      AUTOEXEC.BAT, CONFIG.SYS AND BIOS
      DAUGHTER CARDS
      VERY OLD PRINTER PORTS WITHOUT PULLUPS (PRE 1986)
      IF THE LED IS CYCLING R-Y-G THE KERNEL HAS LOADED
-----
The LPTx name or handle for a port address depends on the operating
system and installed drivers. The DSK uses standard port conventions so
you might need to use a different port name to get the correct port address.
For reference, the systems LPT cross reference table is given below

SYSTEM TABLE LOCATED AT   LPT1 @0x378
RAM ADDRESS 0000:0400      LPT2 @0x278
                           LPT3 @0x002
```

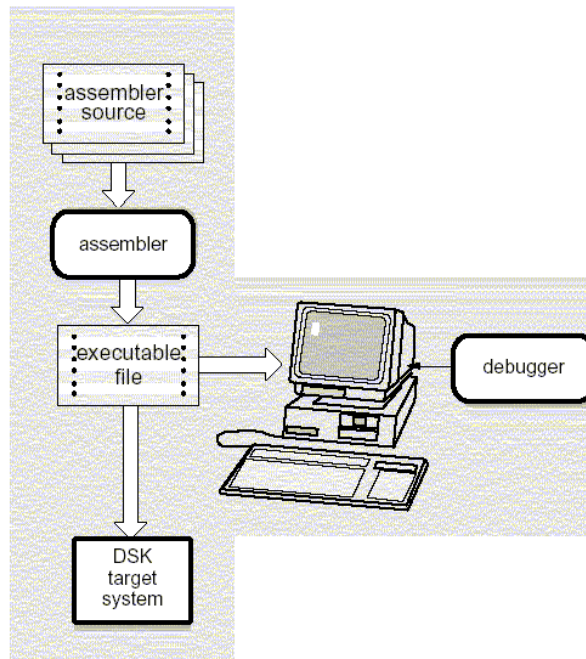
Σχ.Α.4 Κακή εγκατάσταση debugger

Αν, παρόλ' αυτά, ξανά εγκαθιστώντας το λογισμικό, το πρόβλημα παραμένει, ελέγχουμε αν :

- επιλέξαμε λάθος τύπο επικοινωνίας (LPT1 αντί LPT2) με την παράλληλη θύρα
- τη θύρα, με την οποία προσπαθούμε να συνδέσουμε το kit, τη χρησιμοποιεί άλλος πόρος, π.χ. εκτυπωτής
- το καλώδιο της παράλληλης δεν κάνει καλή επαφή
- ο μετασχηματιστής δεν έχει συνδεθεί σωστά. Αν το DSK έχει τροφοδοσία, τότε το LED που έχει, θα πρέπει να είναι κόκκινο ή πράσινο.
- κάναμε εγκατάσταση σε Windows NT/Win2000.

3. Σύντομη περιγραφή της ανάπτυξης κώδικα στο DSK.

Στην παρούσα ενότητα θα δούμε μία γενική περιγραφή του assembler και του debugger. Πρόκειται για το λογισμικό που παρέχεται μαζί με το αναπτυξιακό και που μας βοηθά στην ανάπτυξη και τον έλεγχο κώδικα, υλοποιημένου σε γλώσσα assembly. Η θέση των εργαλείων αυτών στο σύστημα που περιέχει το DSK φαίνεται στο Σχ.Α.5, σχήμα που περιγράφει τη ροή των εργασιών που εκτελούνται κατά την ανάπτυξη κώδικα.



Σχ.Α.5 Ανάπτυξη κώδικα στο DSK

3.1 Περιγραφή του DSK Assembler

Πρόκειται για ένα εργαλείο απλό και εύκολο στη χρήση. Μεταφράζει τον πηγαίο κώδικα, που είναι γραμμένος σε γλώσσα assembly σε object αρχεία γλώσσας μηχανής, κατάλληλα για την TMS320C3x οικογένεια επεξεργαστών. Δεν πρόκειται για έναν COFF assembler, παρόλο που στο αναπτυξιακό θα φορτωθούν και θα τρέξουν εκτελέσιμα object αρχεία, τα οποία παράγονται από τα assembly εργαλεία του DSP.

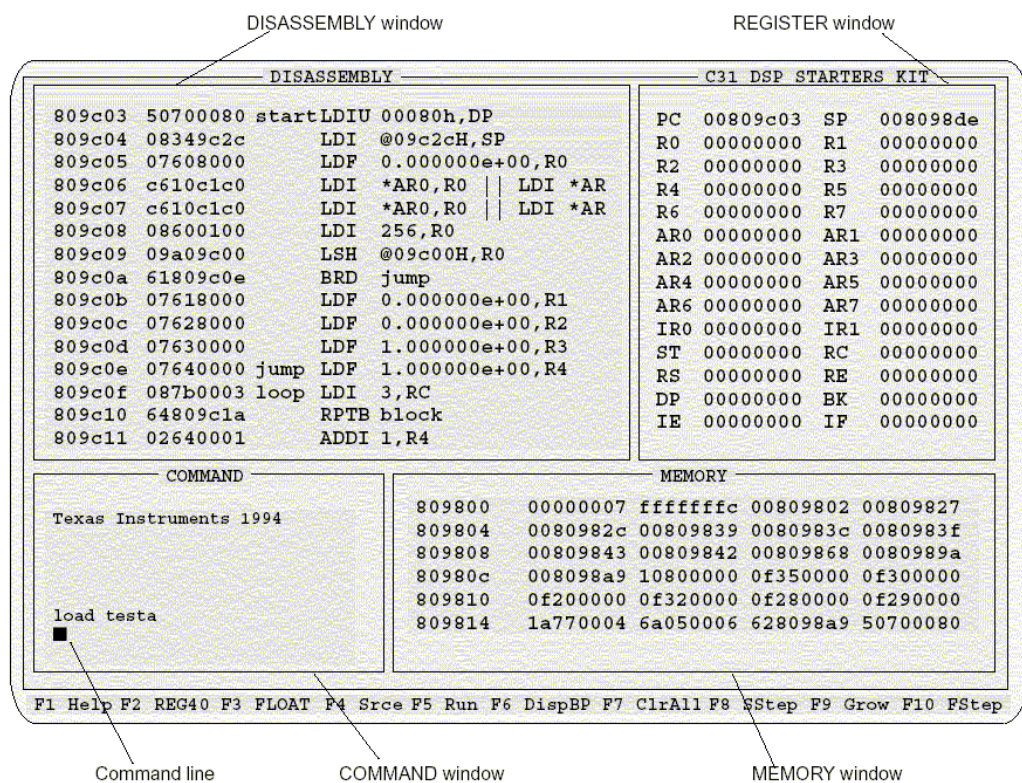
Ο assembler του αναπτυξιακού που περιγράφουμε διαφέρει από άλλους assemblers στο γεγονός ότι δεν περνά από linker για να μπορέσει να παραχθεί το αρχείο εξόδου (output file). Απ'εναντίας, χρησιμοποιεί ειδικές ντιρεκτίβες (directives) για να γίνει σωστά το assembling του κώδικα σε συγκεκριμένη διεύθυνση. Αυτό έχει σαν ως αποτέλεσμα τη δημιουργία μικρών προγραμμάτων γρήγορα κι εύκολα. Εάν, πάλι, θέλουμε να

δημιουργήσουμε μεγαλύτερα προγράμματα, το κάνουμε φτιάχνοντας αλυσιδωτά αρχεία, με τη βοήθεια της ντιρεκτίβας `.include`.

3.2 Περιγραφή του DSK Debugger

Ο debugger του DSK είναι ένα φιλικό εργαλείο, βασισμένο σε περιβάλλον παράθυρων, πράγμα που το κάνει εύχρηστο. Θα παρατηρήσει κανείς ότι δεν είναι απαραίτητη η απομνημόνευση σύνθετων εντολών, ώστε να μπορέσει να το χειριστεί. Παρέχει τη δυνατότητα της single-step (βήμα-βήμα) εκτέλεσης κώδικα, καθώς, επίσης, επιτρέπει και τη χρήση breakpoints.

Στο Σχ.Α.6, βλέπουμε μια μορφή του debugger, όταν αυτός είναι ενεργός. Κάτι παρόμοιο πρέπει να βλέπουμε κάθε φορά που τον χρησιμοποιούμε.



Σχ.Α.6 Βασική οθόνη του Debugger

Βασικό του χαρακτηριστικό αποτελεί το γεγονός ότι εμφανίζει σε ξεχωριστά παράθυρα τον κώδικα, τα δεδομένα, τις θέσεις της μνήμης, ενώ η γραμμή εντολών είναι κι αυτή ευδιάκριτη. Υποστηρίζει ένα μικρό σύνολο εντολών για τη λειτουργία του, σύνολο το οποίο, παράλληλα είναι δυναμικό κι αποτελεσματικό. Έτσι, δεν χρειάζεται κανείς να αποστηθίσει πολλές και δύσκολες εντολές. Τον χρησιμοποιούμε για να ελέγξουμε και να διορθώσουμε κώδικα.

Σκοπός των δύο εργαλείων που περιγράψαμε παραπάνω είναι να δημιουργήσουμε ένα module που θα μπορεί να εκτελεστεί στο τελικό σύστημα του DSK. Συνοπτικά, η διαδικασία του να τρέξουμε ένα πρόγραμμα στο αναπτυξιακό στο οποίο αναφερόμαστε, είναι η ακόλουθη :

1. Δημιουργούμε ένα πηγαίο αρχείο, για παράδειγμα το rand.asm.
2. Εισάγουμε την ακόλουθη εντολή στο command prompt:

```
dsk3a rand.asm <ENTER>
```

Η παραπάνω εντολή ενεργοποιεί τον DSK assembler. Εάν το αρχείο που θέλουμε να τρέξουμε είναι γραμμένο σε assembly, τότε δεν είναι απαραίτητη στην πιο πάνω εντολή να γράψουμε την κατάληξη .asm. Ο assembler την χρησιμοποιεί ως default. Μόλις πατήσουμε <ENTER>, ο assembler παράγει το εκτελέσιμο αρχείο rand.dsk, στο ίδιο directory που περιέχει και το rand.asm. Αυτό είναι το αρχείο που φορτώνουμε και τρέχουμε στο DSK. Επιπλέον, περιέχει και μία λίστα με όλα τα πιθανά λάθη και προειδοποιήσεις (warnings) που μπορεί να έχουν προκύψει. Πέραν αυτού, περιέχει και μία λίστα με τα opcodes του κώδικα.

3. Πλέον, είμαστε έτοιμοι να προχωρήσουμε στο debugging του κώδικα. Εισάγουμε την ακόλουθη εντολή:

```
dsk3d <ENTER>
```

Έχει τεθεί σε λειτουργία ο debugger.

4. Από δω και πέρα, μπορούμε να φορτώσουμε το rand.dsk στο DSK, με τη βοήθεια της εντολής LOAD.

4. Χρησιμοποιώντας τον DSK Assembler.

Για να δημιουργήσουμε πηγαίο κώδικα, τον οποίο θα περάσουμε από τον DSK Assembler, μπορούμε να χρησιμοποιήσουμε σχεδόν κάθε ASCII editor. Τα προγράμματα που είναι γραμμένα στην DSK assembly γλώσσα, μπορούν να περιέχουν ντιρεκτίβες (directives) και εντολές της γλώσσας αυτής, καθώς και σχόλια. Κάθε γραμμή κώδικα μπορεί να περιέχει μέχρι και 80 χαρακτήρες και αποτελείται από το πολύ τέσσερα πεδία. Η γενική της σύνταξη είναι η ακόλουθη:

```
[label] [:] mnemonic [operand list] [:comment]
```

Μερικές, σημαντικές, παρατηρήσεις για τη δημιουργία κώδικα σε γλώσσα DSK Assembly, είναι οι:

- ο Όλες οι δηλώσεις πρέπει να ξεκινούν με ένα label, ή ένα κενό, ή έναν αστερίσκο, ή με ένα ;.

- ο Τα labels είναι προαιρετικά. Αν, όμως, υπάρχουν στον κώδικα, πρέπει να τοποθετούνται στην πρώτη στήλη αυτού. Μπορούν να περιέχουν μέχρι οκτώ αλφαριθμητικούς χαρακτήρες, είναι case-sensitive και δεν μπορεί ο πρώτος τους χαρακτήρας να είναι αριθμός.

- ο Κάθε ένα από τα πεδία της παραπάνω σύνταξης, πρέπει να διαχωρίζονται μεταξύ τους από ένα ή περισσότερα κενά.

- ο Τα σχόλια είναι προαιρετικά. Τα σχόλια που ξεκινούν από την πρώτη στήλη του κώδικα, μπορούν να ξεκινούν με * ή ;, ενώ τα σχόλια που ξεκινούν από οποιαδήποτε άλλη στήλη, πρέπει να ξεκινούν με ;.

- ο Το πεδίο mnemonic της σύνταξης που αναφέρθηκε παραπάνω, μπορεί να περιέχει είτε εντολές ή directives της γλώσσας assembly.

- ο Το operand field περιέχει τελεστές. Αυτοί μπορεί να είναι είτε σταθερές ή σύμβολα ή συνδυασμός αυτών. Διαχωρίζονται μεταξύ τους με κόμματα.

Είναι αξιοσημείωτο ότι ο DSK Assembler περιλαμβάνει αρκετές directives, οι οποίες ελέγχουν την αρχική διεύθυνση των διαφόρων τμημάτων του κώδικα. Έτσι δεν είναι απαραίτητος κάποιος linker. Για παράδειγμα, μπορούμε να δημιουργήσουμε ένα αρχείο που να περιλαμβάνει όλες τις αρχικές διευθύνσεις των τμημάτων του εκάστοτε κώδικα, το οποίο, με τη χρησιμοποίηση της ντιρεκτίβας `.include`, να το προσάπτουμε σε όποιο αρχείο είναι απαραίτητο.

Πριν, λοιπόν, επιχειρήσουμε να περάσουμε τα προγράμματά μας από τον debugger, πρέπει να τα περάσουμε από τον assembler του συστήματος, δίνοντας την εντολή:

`dsk3a filenames,`

όπου στη θέση *filenames* μπορούν να μπει ένα ή περισσότερα αρχεία, τα οποία δεν είναι case-sensitive. Εάν, σε κάποιο από αυτά δεν βάλουμε κατάληξη, ο assembler υποθέτει ότι έχει την κατάληξη `.asm`.

5. Χρησιμοποιώντας τον DSK Debugger.

Έχοντας εξασφαλίσει τη σωστή λειτουργία του όλου συστήματος, μπορούμε να χειριστούμε τον debugger που διαθέτει. Στη συνέχεια, θα δούμε διεξοδικά όλα τα μέρη-παράθυρα που τον απαρτίζουν, με σκοπό την πλήρη κατανόησή του.

A) Παράθυρο Disassembly

Το παράθυρο αυτό μας δείχνει τα περιεχόμενα της μνήμης κατά τη διαδικασία του assembling. Όπως φαίνεται και στο Σχ.Α.7, το παράθυρο περιέχει αρκετές γραμμές κώδικα. Κάθε γραμμή κώδικα, εκτός της εντολής, περιλαμβάνει τη διεύθυνση αυτής, το opcode της και το label που μπορεί να προηγείται της εντολής.

Instruction address	Instruction opcode	Label	Instruction mnemonic
DISASSEMBLY			
809c03	50700080	start	LDIU 00080h, DP
809c04	08349c2c		LDI @09c2cH, SP
809c05	07608000		LDF 0.000000e+00, R0
809c06	c610c1c0		LDI *AR0, R0 LDI *AR
809c07	c610c1c0		LDI *AR0, R0 LDI *AR
809c08	08600100		LDI 256, R0
809c09	09a09c00		LSH @09c00H, R0
809c0a	61809c0e		BRD jump
809c0b	07618000		LDF 0.000000e+00, R1
809c0c	07628000		LDF 0.000000e+00, R2
809c0d	07630000		LDF 1.000000e+00, R3
809c0e	07640000	jump	LDF 1.000000e+00, R4
809c0f	087b0003	loop	LDI 3, RC
809c10	64809c1a		RPTB block
809c11	02640001		ADDI 1, R4

Σχ.Α.7 Παράθυρο Disassembly του DSK debugger

Κάθε φορά στο παράθυρο, είναι “επιλεγμένη” κάποια εντολή. Αυτό δηλώνει την επόμενη εντολή που πρόκειται να εκτελεστεί. Για να επιλέξουμε το παράθυρο Disassembly, πατάμε <ALT> + <D>. Έχουμε τη δυνατότητα να επιλέξουμε μία γραμμή κώδικα και να ορίσουμε σε αυτή breakpoint.

B) Παράθυρο των CPU REGISTER

Εμφανίζει τα περιεχόμενα όλων των CPU Registers, όπως φαίνεται και στο Σχ.Α.8. Αρχικά, τα περιεχόμενα των καταχωρητών εμφανίζονται σε δεκαεξαδική μορφή. Μπορούμε να δούμε τους καταχωρητές εκτεταμένης ακρίβειας (extended – precision registers) σε δεκαδική μορφή κινητής υποδιαστολής πατώντας <F3>, ενώ για να τους δούμε σε 40-bit δεκαεξαδική μορφή, πατάμε <F2>.

Για να ενεργοποιήσουμε το παράθυρο που περιγράφουμε, πατάμε <ALT> + <C>. Έχουμε τη δυνατότητα να αλλάξουμε τα περιεχόμενα κάποιου καταχωρητή, επιλέγοντάς τον, τυπώνοντας τη νέα τιμή και πατώντας <ENTER>, ώστε να επικυρωθεί η αλλαγή.

Ονόματα καταχωρητών

C31 DSP STARTERS KIT			
PC	00809c03	SP	008098de
R0	00000000	R1	00000000
R2	00000000	R3	00000000
R4	00000000	R5	00000000
R6	00000000	R7	00000000
AR0	00000000	AR1	00000000
AR2	00000000	AR3	00000000
AR4	00000000	AR5	00000000
AR6	00000000	AR7	00000000
IR0	00000000	IR1	00000000
ST	00000000	RC	00000000
RS	00000000	RE	00000000
DP	00000000	BK	00000000
IE	00000000	IF	00000000

Περιεχόμενα καταχωρητών

Σχ.Α.8 Παράθυρο των CPU Registers

Γ) Παράθυρο Μνήμης (Memory Window)

Παρουσιάζει τα περιεχόμενα των θέσεων μνήμης. Όπως δείχνει και το Σχ.Α.9, αποτελείται από δύο μέρη:

- Διευθύνσεις: Η πρώτη στήλη με αριθμούς ορίζει τις διευθύνσεις της πρώτης στήλης των δεδομένων μνήμης.
- Δεδομένα: Οι υπόλοιπες στήλες που φαίνονται στο Memory Window, είναι οι τιμές που περιέχονται στις αντίστοιχες διευθύνσεις.

Στήλη διευθύνσεων

Στήλη δεδομένων

MEMORY				
809800	00000007	fffffffc	00809802	00809827
809804	0080982c	00809839	0080983c	0080983f
809808	00809843	00809842	00809868	0080989a
80980c	008098a9	10800000	0f350000	0f300000
809810	0f200000	0f320000	0f280000	0f290000
809814	1a770004	6a050006	628098a9	50700080

Σχ.Α.9 Memory Window του debugger

Για παράδειγμα, το παραπάνω σχήμα έχει τέσσερις στήλες δεδομένων, οπότε κάθε διεύθυνση στη στήλη διευθύνσεων αυξάνεται κατά 4. Έτσι, η διεύθυνση 0x0080 9800

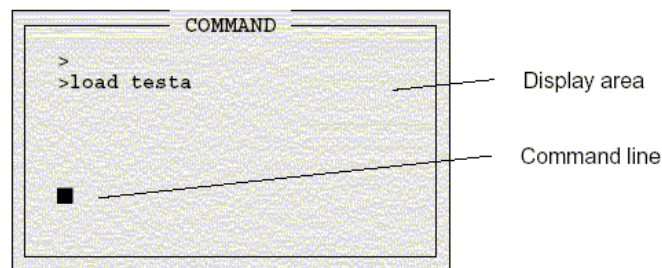
περιέχει την τιμή 0x0000 0007, η διεύθυνση 0x0080 9801 περιέχει την τιμή 0xFFFF FFFC, η διεύθυνση 0x0080 9807 περιέχει την τιμή 0x0080 983F, κ.ο.κ.

Για να ενεργοποιήσουμε το παράθυρο της μνήμης, πατάμε <ALT> + <M>. Και εδώ, μπορούμε να αλλάξουμε τα δεδομένα που περιέχει μία διεύθυνση, με τον ίδιο τρόπο, όπως και στο παράθυρο των καταχωρητών.

4) Παράθυρο εντολών (Command Window)

Είναι η περιοχή του debugger, όπου εισάγουμε εντολές και βλέπουμε τα μηνύματα λάθους. Φαίνεται μεμονωμένο στο Σχ.Α.10. Αποτελείται από δύο μέρη:

- Command Line: Είναι η περιοχή όπου εισάγουμε τις εντολές. Για να το κάνουμε αυτό, απλά, γράφουμε την εντολή που θέλουμε, ανεξάρτητα με το ποιο παράθυρο του debugger είναι ενεργό εκείνη τη στιγμή.
- Display Area: Αυτή η περιοχή εμφανίζει τις εντολές που έχουμε εισάγει, καθώς και τα μηνύματα λάθους του debugger.



Σχ.Α.10 Command Window του debugger

6. Εντολές του Debugger.

Οι παρακάτω πίνακες περιέχουν περίληψη των σημαντικότερων εντολών του DSK Debugger.

Εντολή	Περιγραφή
SS	Τρέχει μία εντολή τη φορά (Single-Step)
XN <i>n</i>	Εκτελεί <i>n</i> εντολές
XG <i>addr</i>	Εκτελεί μία-μία τις εντολές,

	μέχρι να φτάσει στη διεύθυνση <i>addr</i>
RUN	Εκτελεί το πρόγραμμα, μέχρι να συναντήσει breakpoint
RUNF	Εκτελεί το πρόγραμμα και αγνοεί τα breakpoints (Run-Free)

Πίνακας Α.1 Εντολές εκτέλεσης Προγράμματος

Εντολή	Περιγραφή
MEM <i>addr</i>	Εμφανίζει τα περιεχόμενα της μνήμης στο Memory Window, αρχίζοντας από τη διεύθυνση <i>addr</i>
MM <i>addr</i>	Τροποποιεί τα δεδομένα της μνήμης στη θέση <i>addr</i>
MM <i>addr leng val</i>	Γεμίζει <i>leng</i> θέσεις μνήμης με την τιμή <i>val</i> , αρχίζοντας από τη θέση <i>addr</i>
DASM <i>addr</i>	Εμφανίζει στο Disassembly Window τον assembly κώδικα, ξεκινώντας από τη διεύθυνση <i>addr</i>
REG40	Εμφανίζει τους καταχωρητές εκτεταμένης ακρίβειας σε 40-bit δεκαεξαδική μορφή, στο Παράθυρο Καταχωρητών
FLOAT	Εμφανίζει τους καταχωρητές εκτεταμένης ακρίβειας σε δεκαδική μορφή κινητής υποδιαστολής, στο Παράθυρο Καταχωρητών

Πίνακας Α.2 Εμφάνιση και τροποποίηση δεδομένων

Εντολή	Περιγραφή
SB <i>addr</i>	Ορίζει breakpoint στη διεύθυνση <i>addr</i>
CB <i>addr</i>	Καταργεί breakpoint από τη

	διεύθυνση <i>addr</i>
CB	Καταργεί όλα τα breakpoints
DB	Εμφανίζει λίστα με όλα τα breakpoints, που έχουν τεθεί

Πίνακας Α.3 Διαχείριση Breakpoints

Εντολή	Περιγραφή
LOAD <i>filename</i>	Φορτώνει ένα object αρχείο στο DSK
SLOAD <i>filename</i>	Φορτώνει σύμβολα
BLOAD <i>filename</i>	Φορτώνει μόνο το Binary
SCLEAR	Καταργεί σύμβολα

Πίνακας Α.4 Διαχείριση Προγραμμάτων

Εντολή	Περιγραφή
RESET	Εκτελεί RESET στο DSK
QUIT or EXIT	Κλείνει τον debugger
DOS (<i>expression to Run</i>)	Μπαίνει σε DOS περιβάλλον και εκτελεί την <i>expression</i> . Πατώντας EXIT, επιστρέφει στον debugger
EDIT <i>filename</i>	Μπαίνει σε DOS περιβάλλον και εκτελεί τον editor για το <i>filename</i> . Αν δεν δοθεί <i>filename</i> , χρησιμοποιεί το όνομα του τελευταίου αρχείου που έχει φορτώσει.
dsk3a <i>filename.asm</i>	Μπαίνει σε DOS περιβάλλον και εκτελεί τον DSK assembler για το <i>filename</i>

Πίνακας Α.5 Διαχείριση του Συστήματος

7. Το TLC32040 AIC.

Όπως έχουμε, ήδη, αναφέρει, το H/W και το S/W του TMS320C3x DSK, λειτουργούν έτσι ώστε να μας παρέχουν μία αναπτυξιακή πλατφόρμα χαμηλού κόστους, η οποία μας επιτρέπει την ανάπτυξη εφαρμογών πραγματικού χρόνου.

Το 'C3x DSK ξεκινά τη λειτουργία του αντιδρώντας στην εντολή RESET του Host PC και φορτώνοντας τον πυρήνα επικοινωνίας που του παρέχουμε. Ο πυρήνας αυτός περιέχει όλα τα απαραίτητα I/O για τη σωστή επικοινωνία μεταξύ του αναπτυξιακού και του Host PC μέσω της παράλληλης θύρας του 'C31. Τα αναλογικά I/O στέλνονται στη σειριακή θύρα του 'C31 από το TLC32040 (AIC).

Το AIC του αναπτυξιακού παρέχει:

- Αναλογικό interface εισόδου-εξόδου, χάρη στους 14-bit ADC και DAC που διαθέτει. Ο ρυθμός δειγματοληψίας τους φτάνει μέχρι και τα 20,000 δείγματα το δευτερόλεπτο.
- Φίλτρο εξόδου (επανακατασκευής σημάτων)
- Φίλτρο εισόδου για την αντιμετώπιση φαινομένων aliasing. Δεν είναι απαραίτητη η χρήση του για κάθε σήμα εισόδου.
- Βοηθητικό κανάλι αναλογικής εισόδου.

Το TLC32040 συνδέεται με τη σειριακή θύρα του 'C31 μέσω ενός header. Έτσι, είναι δυνατή η αποσύνδεση του AIC και η χρησιμοποίηση της σειριακής θύρας του 'C31 για τη επικοινωνία του αναπτυξιακού με άλλη κάρτα.

Το ρολόι του AIC, καθώς και το σήμα για το Reset αυτού ελέγχονται από δύο pins του 'C31, το TIMER0 pin, που οδηγεί το master clock του 'C31 στο AIC, και το σήμα XF0, το οποίο είναι υπεύθυνο για το Reset του AIC.

BIBΛΙΟΓΡΑΦΙΑ

Βιβλία & Papers:

TMS320C3x User's Guide, SPRU031D, Texas Instruments.

TMS320C3x DSP Starter Kit, SPRU163, Texas Instruments.

TMS320C3x General-Purpose Applications, SPRU194, Texas Instruments.

Using TMS320 Family DSPs in Motion Control Systems, Cotistatm Ilas, Aurelian Sarca, Radu Giuclea, Liviu Kreindler, ESIEE Paris, September 1996, SPRA327.

Designing a DSP System, App. Note, D. SignT.

Precision Digital Sine-Wave Generation with the TMS32010, SPRA007, Texas Instruments.

TMS320 Algirithm Debugging Technics, SPRA084, Texas Instruments.

TMS320 Hardware Applications, SPRA119, Texas Instruments.

Teaching the TMS320C3x DSP as Individual Project, M. Otesteanu, ESIEE, Paris, September 1996, SPRA343.

TMS320C31, TMS320LC31 Digital Signal Processors, SPRS035B.

TL081, TL081A, TL081B, TL082, TL082A, TL082B, TL082Y, TL084, TL084A, TL084B, TL084Y, JFET-INPUT OPERATIONAL AMPLIFIERS, SLOS081S, FEBRUARY 1999.

ΗΛΕΚΤΡΟΝΙΚΗ, Malvino, 5^η έκδοση.

Power Electronics, Mohan & Undeland & Robbins, John Wiley & Sons Inc.

Multilevel PWM Methods at Low Modulation Indices, Leon M. Tolbert, Fang Z. Peng, Thomas G. Habetler.

Inverter Design for 2001 Future Energy Challenge, Un. Of Central Florida, Orlando, June 2001.

A Dual-Stage PWM DC to AC Inverter with reduced Harmonic Distortion and Switching Losses, Department of Electrical & Electronic Engineering, Un. Of Mauritius, September 1999.

Three-Phase Sine-Wave Generation using the PWM Unit of the ADMC401, AN401-03, Analog Devices.

Power Electronics, A.M Gole, 2000.

A New Bidirectional, Sinusoidal, High-Frequency Inverter Design, Eftichios Koutroulis, John Chatzakis, Kostas Kalaitzakis, Nickolas C. Voulgaris.

Ηλεκτρονικά Ισχύος, Στέφανος Μανιάς.

Filter Circuits, ECEN 2260, Oct. 2001.

Butterworth Filters, Adam Blair, Richard Baraniuk, 2001.

Wind Energy Conversion Systems, Liuchen Chang, Spring 2002.