

16.9 Υλοποίηση Συστημάτων Διακριτού Χρόνου

Ως τώρα έχουμε συναντήσει τρεις διαφορετικές περιγραφές ενός ΓΧΑ συστήματος:

- την εξίσωση διαφορών εισόδου-εξόδου (με σταθερούς συντελεστές) του
- την κρουστική απόκρισή του $h[n]$
- τη συνάρτηση μεταφοράς του $H(z)$

Και οι τρεις αυτές περιγραφές είναι θεωρητικά ισοδύναμες. Όταν όμως υλοποιούμε στην πράξη τέτοια συστήματα, δηλ. με αναλογικά ή ψηφιακά κυκλώματα, η εξίσωση διαφορών ή η συνάρτηση μεταφοράς πρέπει να μετατραπεί σε έναν αλγόριθμο ή σε μια δομή που να μπορεί να πραγματοποιηθεί με την υπάρχουσα τεχνολογία. Όπως θα δούμε σε αυτό το κεφάλαιο, βασικοί δομικοί λίθοι που χρησιμοποιούνται σε αυτές τις δομές είναι οι βασικές πράξεις της πρόσθεσης, του πολλαπλασιασμού με μια σταθερά, και της καθυστέρησης (αποθήκευσης στη μνήμη).

Ας δούμε ένα απλό παράδειγμα. Έστω το σύστημα με συνάρτηση μεταφοράς

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 - a z^{-1}}, \quad |z| > |a| \quad (16.422)$$

Η κρουστική απόκριση του παραπάνω συστήματος δίνεται εύκολα (έτσι δεν είναι; :-) ως:

$$h[n] = b_0 a^n u[n] + b_1 a^{n-1} u[n-1] \quad (16.423)$$

και η πρώτη τάξης εξίσωση διαφορών δίνεται ως

$$y[n] - a y[n-1] = b_0 x[n] + b_1 x[n-1] \quad (16.424)$$

Αν θέλουμε να υλοποιήσουμε αυτό το σύστημα, η πράξη της συνέλιξης δε θα ήταν βολική, γιατί η κρουστική απόκριση $h[n]$ είναι άπειρη σε διάρκεια, οπότε θα χρειαζόμασταν άπειρες θέσεις μνήμης για να αποθηκεύσουμε την κρουστική απόκριση ή το αποτέλεσμα της συνέλιξης.

Όμως μπορούμε να γράψουμε την εξίσωση διαφορών ως

$$y[n] = a y[n-1] + b_0 x[n] + b_1 x[n-1] \quad (16.425)$$

η οποία μας δίνει τη βάση ενός αναδρομικού αλγορίθμου για τον υπολογισμό της εξόδου οποιαδήποτε χρονική στιγμή n , συναρτήσει της προηγούμενης εξόδου $y[n-1]$ και των εισόδων $x[n]$ και $x[n-1]$. Όπως γνωρίζετε, αν θεωρήσουμε συνθήκες αρχικής ηρεμίας στο σύστημα (δηλ. αν $x[n] = 0$, $n < 0$, τότε $y[n] = 0$, $n < 0$) και χρησιμοποιήσουμε τη Σχέση (16.425) ως αναδρομική εξίσωση για τον υπολογισμό της εξόδου, το σύστημα θα είναι

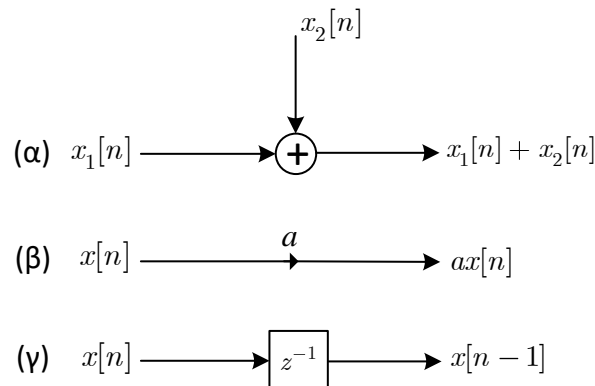
τότε γραμμικό και χρονικά αμετάβλητο. Μπορούμε να γενικεύσουμε το πρόβλημά μας σε συστήματα N τάξης, αλλά είτε το παραπάνω σύστημα είτε η γενίκευσή του σε N τάξη δεν αποτελούν το μοναδικό αλγόριθμο υπολογισμού ενός συστήματος. Θα δούμε ότι υπάρχουν πολλές επιλογές για την υλοποίηση ενός συστήματος που έχει μια δεδομένη εξίσωση διαφορών. Η διαφορά ανάμεσα σε αυτές τις επιλογές συναντώνται

1. στο πλήθος των πράξεων
2. στο πλήθος των θέσεων μνήμης (καθυστέρηση)
3. σε άλλες παραμέτρους που δεν εξετάζουμε εδώ (ευρωστία στην κβαντοποίηση, κλπ.)

16.9.1 Αναπαράσταση Εξισώσεων Διαφορών με Διάγραμμα

Η υλοποίηση ενός ΓΧΑ συστήματος με χρήση αναδρομικού αλγόριθμου ο οποίος προέρχεται από την εξίσωση διαφορών απαιτεί προηγούμενες τιμές της εισόδου ή της εξόδου, όπως φαίνεται και στη Σχέση (16.425). Η καθυστέρηση των τιμών μιας ακολουθίας υπονοεί ότι υπάρχει ανάγκη αποθήκευσής τους σε μια μνήμη. Επίσης, πρέπει να έχουμε μηχανισμούς πολλαπλασιασμού τιμών με σταθερές, όπως και μηχανισμούς πρόσθεσης τιμών.

Με βάση τα παραπάνω, τα βασικά δομικά στοιχεία για την υλοποίηση ενός ΓΧΑ συστήματος είναι οι *αθροιστές*, οι *πολλαπλασιαστές*, και οι *θέσεις μνήμης*. Μια αναπαράστασή τους φαίνεται στο Σχήμα (16.45). Μια μικρή



Σχήμα 16.45: Διαγράμματα για (α) πρόσθεση ακολουθιών, (β) πολλαπλασιασμό ακολουθίας με σταθερά, (γ) μοναδιαία καθυστέρηση.

μόνο παρατήρηση για τη μοναδιαία καθυστέρηση, καθώς τα υπόλοιπα διαγράμματα είναι προφανή. Σε ψηφιακές υλοποιήσεις, η καθυστέρηση μπορεί να υλοποιηθεί ως ένας καταχωρητής για κάθε μοναδιαία καθυστέρηση. Σε αναλογικές υλοποιήσεις, η καθυστέρηση μπορεί να υλοποιηθεί ως συσκευή αποθήκευσης φορτίου (όμοια με τους πυκνωτές). Στα διαγράμματα που θα χρησιμοποιήσουμε, θα συμβολίσουμε την καθυστέρηση με τον όρο z^{-1} , όπως στο Σχήμα.

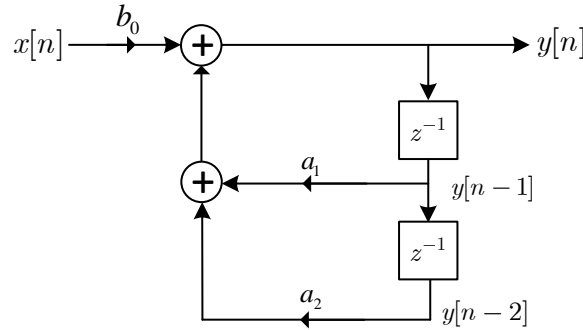
Ας προσπαθήσουμε λοιπόν τώρα να κατασκευάσουμε ένα διάγραμμα για το σύστημα που περιγράφεται ως

$$y[n] - a_1 y[n-1] - a_2 y[n-2] = b_0 x[n] \quad (16.426)$$

του οποίου η συνάρτηση μεταφοράς του δίνεται ως

$$H(z) = \frac{b_0}{1 - a_1 z^{-1} - a_2 z^{-2}} \quad (16.427)$$

Το διάγραμμα που προκύπτει φαίνεται στο Σχήμα 16.46. Όταν το σύστημα υλοποιείται είτε σε έναν υπολογιστή είτε σε ένα chip επεξεργασίας σήματος, δομές όπως αυτές του Σχήματος 16.46 χρησιμοποιούνται σαν βάσεις για το πρόγραμμα που υλοποιεί το σύστημα. Αν το σύστημα υλοποιείται με VLSI τεχνολογία, το διάγραμμα είναι η βάση για τον ορισμό της αρχιτεκτονικής του συστήματος, από πλευράς υλικού. Σε κάθε περίπτωση, τέτοια διαγράμματα δείχνουν ρητά ότι πρέπει να χρησιμοποιηθεί μνήμη για την αποθήκευση μεταβλητών (στην περίπτωσή μας, για τα $y[n-1]$, $y[n-2]$). Επίσης, βλέπουμε από το Σχήμα 16.46 ότι η έξοδος $y[n]$ υπολογίζεται δημιουργώντας πρώτα το γινόμενο των $a_1 y[n-1]$ και $a_2 y[n-2]$, μετά προσθέτοντάς τα μεταξύ τους, και τέλος, προσθέτοντας το αποτέλεσμα στο $b_0 x[n]$. Άρα, το διάγραμμα αναπαριστά με βολικό τρόπο την πολυπλοκότητα που σχετίζεται με τον αλγόριθμο υπολογισμού, τα βήματα του αλγόριθμου, και το ποσό του υλικού που απαιτείται για την πραγματοποίηση του συστήματος!



Σχήμα 16.46: Παράδειγμα διαγράμματος για την εξίσωση διαφορών της Σχέσης (16.426).

Ας γενικεύσουμε το παράδειγμά μας σε μεγαλύτερης τάξης εξίσωση διαφορών, ως

$$y[n] - \sum_{k=1}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k] \tag{16.428}$$

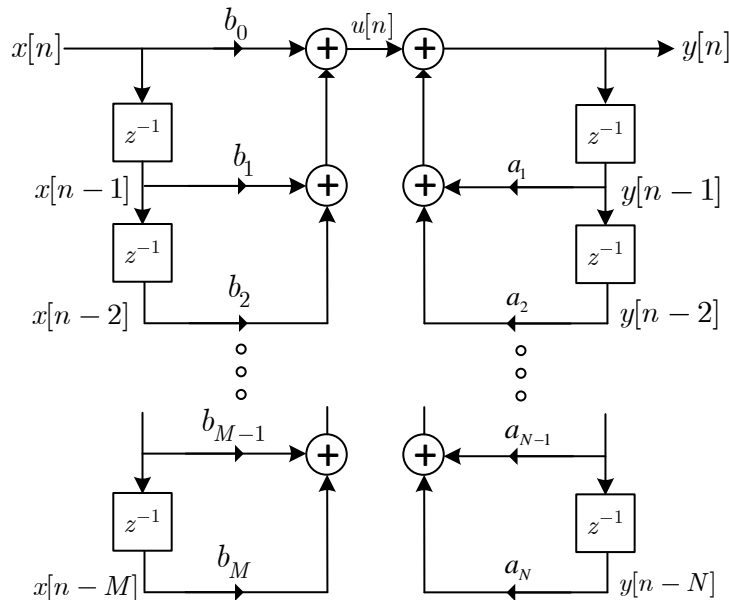
η οποία έχει συνάρτηση μεταφοράς

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 - \sum_{k=1}^N a_k z^{-k}} \tag{16.429}$$

Ξαναγράφοντας τη Σχέση (16.428) ως αναδρομικό τύπο υπολογισμού της εξόδου, θα έχουμε

$$y[n] = \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^M b_k x[n-k] \tag{16.430}$$

Το διάγραμμα που αναπαριστά αυτόν τον τρόπο υπολογισμού της εξόδου του συστήματος φαίνεται στο Σχήμα 16.47. Το διάγραμμα του Σχήματος 16.47 είναι μια ρητή απεικόνιση της Σχέσης (16.430). Συγκεκριμένα, αναπαριστά το



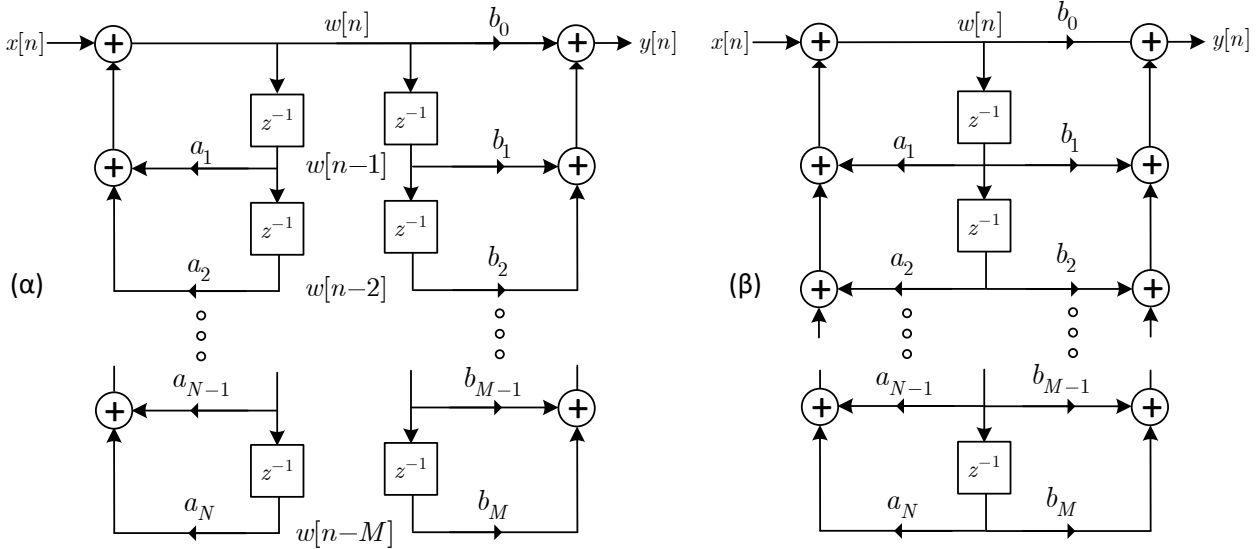
Σχήμα 16.47: Παράδειγμα διαγράμματος για την εξίσωση διαφορών της Σχέσης (16.428).

ζεύγος εξισώσεων διαφορών

$$v[n] = \sum_{k=0}^M b_k x[n-k] \tag{16.431}$$

$$y[n] = \sum_{k=1}^N a_k y[n-k] + v[n] \quad (16.432)$$

Ένα διάγραμμα μπορεί να αναδιαταχθεί ή και τροποποιηθεί με πολλούς τρόπους χωρίς να μεταβάλλουμε τη συνάρτηση μεταφοράς που αναπαριστά. Κάθε μετατροπή ή αναδιάταξη αναπαριστά μια διαφορετική υλοποίηση του ίδιου συστήματος, δηλ. έναν διαφορετικό υπολογιστικό αλγόριθμο για το ίδιο σύστημα. Για παράδειγμα, το παράδειγμα του Σχήματος 16.47 μπορεί να ιδωθεί ως δυο συστήματα σε σειρά, με το πρώτο να υπολογίζει το $v[n]$ από το $x[n]$, και το δεύτερο να υπολογίζει το $y[n]$ από το $v[n]$. Δεδομένου ότι καθένα από αυτά τα δυο συστήματα είναι ΓΧΑ (θεωρώντας συνθήκες αρχικής ηρεμίας για τους καταχωρητές), η σειρά με την οποία βρίσκονται τα δυο συστήματα στο διάγραμμα μπορεί να αλλάξει, χωρίς να επηρεάζει αυτό το συνολικό σύστημα! Αυτή η αναδιάταξη φαίνεται στο Σχήμα 16.48(α). Στο Σχήμα 16.48(α) έχουμε υποθέσει ότι η τάξη αριθμητή και παρονομαστή είναι ίδια ($M = N$).



Σχήμα 16.48: Αναδιάταξη διαγράμματος (α) για την εξίσωση διαφορών της Σχέσης (16.428) και (β) ίδιο διάγραμμα με λιγότερες θέσεις μνήμης.

Αν είναι διαφορετική, τότε απλά κάποιοι συντελεστές a_k ή b_k θα απαλειφθούν από το Σχήμα, και οι αντίστοιχοι κλάδοι θα εξαφανιστούν.

Αν δούμε τα δυο αυτά διαγράμματα με όρους συνάρτησης μεταφοράς, το Σχήμα 16.47 μπορεί να ιδωθεί ως η υλοποίηση του $H(z)$ μέσω της γραφής του ως

$$H(z) = H_2(z)H_1(z) = \left(\frac{1}{1 - \sum_{k=1}^N a_k z^{-k}} \right) \sum_{k=0}^M b_k z^{-k} \quad (16.433)$$

ή ισοδύναμα, ως το ζεύγος εξισώσεων

$$V(z) = H_1(z)X(z) = \left(\sum_{k=0}^M b_k z^{-k} \right) X(z) \quad (16.434)$$

$$Y(z) = H_2(z)V(z) = \left(\frac{1}{1 - \sum_{k=1}^N a_k z^{-k}} \right) V(z) \quad (16.435)$$

Από την άλλη μεριά, το διάγραμμα του Σχήματος 16.48(α) μπορεί να ιδωθεί ως η υλοποίηση του $H(z)$ μέσω της γραφής του ως

$$H(z) = H_1(z)H_2(z) = \sum_{k=0}^M b_k z^{-k} \left(\frac{1}{1 - \sum_{k=1}^N a_k z^{-k}} \right) \quad (16.436)$$

ή ισοδύναμα, ως το ζεύγος εξισώσεων

$$W(z) = H_2(z)X(z) = \left(\frac{1}{1 - \sum_{k=1}^N a_k z^{-k}} \right) X(z) \quad (16.437)$$

$$Y(z) = H_1(z)W(z) = \left(\sum_{k=0}^M b_k z^{-k} \right) W(z) \quad (16.438)$$

Τα διαγράμματα των Σχημάτων 16.47, 16.48(α) έχουν αρκετές σημαντικές διαφορές, τις οποίες πρέπει να επισημάνουμε. Στο διάγραμμα 16.47 υλοποιούνται πρώτα τα μηδενικά του $H(z)$, δηλ. το υποσύστημα $H_1(z)$, και μετά οι πόλοι, δηλ. το υποσύστημα $H_2(z)$. Στο διάγραμμα 16.48(β) υλοποιούνται πρώτα οι πόλοι του $H(z)$, δηλ. το υποσύστημα $H_2(z)$, και μετά τα μηδενικά, δηλ. το υποσύστημα $H_1(z)$. Θεωρητικά, η σειρά της υλοποίησης δεν επηρεάζει το συνολικό σύστημα. Όμως, όπως θα δούμε, όταν μια εξίσωση διαφορών υλοποιείται με αριθμητική πεπερασμένης ακρίβειας, μπορεί να υπάρξουν σημαντικές διαφορές ανάμεσα στα δυο συστήματα που είναι θεωρητικά ισοδύναμα. Μια ακόμα σημαντική διαφορά είναι ο αριθμός των καθυστερήσεων στα δυο διαγράμματα. Και στα δυο, απαιτούνται συνολικά $N + M$ στοιχία καθυστέρησης (μνήμη). Όμως, το διάγραμμα του Σχήματος 16.48(α) μπορεί να επανασχεδιαστεί αν προσέξουμε ότι το ίδιο ακριβώς σήμα ($w[n]$) αποθηκεύεται δυο φορές, από μια φορά σε κάθε αλυσίδα υπολογισμών. Κατά συνέπεια, μπορούμε να πετάξουμε τη μια αλυσίδα, χρησιμοποιώντας μόνο μια αλυσίδα θέσεων μνήμης, κοινή ανάμεσα στις δυο αλυσίδες υπολογισμών. Αυτή η παρατήρηση οδηγεί στο διάγραμμα του Σχήματος 16.48(β). Σε αυτό το διάγραμμα, οι συνολικές καθυστερήσεις (θέσεις μνήμης) είναι λιγότερες από αυτές των δυο προηγούμενων διαγραμμάτων. Για την ακρίβεια, οι θέσεις μνήμης τώρα είναι ο ελάχιστος αριθμός θέσεων μνήμης l που απαιτείται για την υλοποίηση ενός συστήματος με συνάρτηση μεταφοράς $H(z)$ όπως στη Σχέση (16.429), και δίνεται εν γένει από τη σχέση

$$l = \max(N, M) \quad (16.439)$$

Μια υλοποίηση η οποία γίνεται με τον ελάχιστο αριθμό στοιχείων καθυστέρησης αναφέρεται συχνά ως υλοποίηση *κανονικής μορφής* - *canonic form*. Η μη κανονική μορφή υλοποίησης που φαίνεται στο Σχήμα 16.47 αναφέρεται ως υλοποίηση Direct Form I ενός γενικού συστήματος τάξης N , επειδή είναι μια ευθεία (direct) υλοποίηση της εξίσωσης διαφορών. Το διάγραμμα του Σχήματος 16.48(β) αναφέρεται συχνά ως υλοποίηση Direct Form II ή Canonic Direct Form.

Ας δούμε ένα μικρό παράδειγμα.

Παράδειγμα 16.32:

Έστω το ΓΧΑ σύστημα που δίνεται από τη συνάρτηση μεταφοράς

$$H(z) = \frac{1 + 3z^{-1}}{1 - \frac{1}{4}z^{-1} + \frac{1}{2}z^{-2}} \quad (16.440)$$

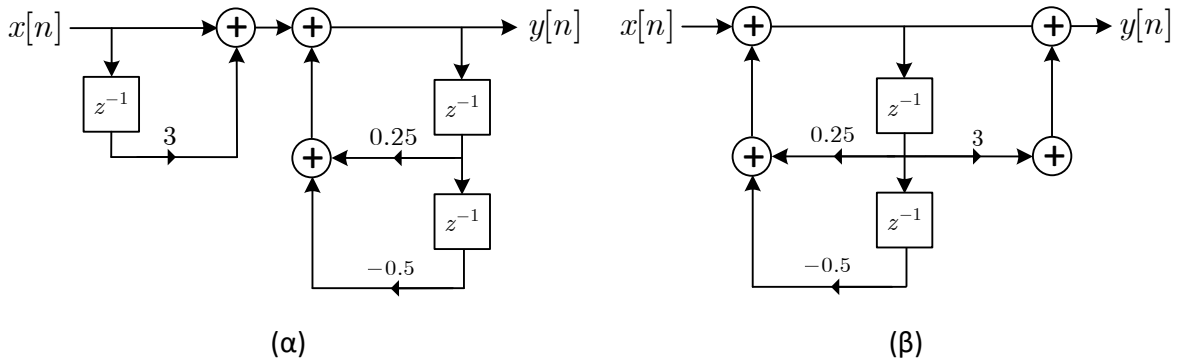
Σχεδιάστε τα Direct Forms I, II.

Λύση:

Συγκρίνοντας την παραπάνω εξίσωση με τη Σχέση (16.429), παρατηρούμε ότι $b_0 = 1$, $b_1 = 3$, $a_1 = \frac{1}{4}$, $a_2 = -\frac{1}{2}$. Έτσι, το Direct Form I και Direct Form II διάγραμμα φαίνεται στο Σχήμα 16.49. Προσέξτε ότι οι συντελεστές του παρονομαστή (πλην της μονάδας του πρώτου συντελεστή) έχουν αντίθετα πρόσημα από αυτά στην συνάρτηση μεταφοράς. Επίσης, προσέξτε ότι η Direct Form II σχεδίαση απαιτεί μόνο δυο καθυστερήσεις, μια λιγότερη από τις τρεις που απαιτεί η υλοποίηση σε Direct Form I. ■

Αυτή λοιπόν είναι μια όψη: της σχεδίασης των διαγραμμάτων με τους παραπάνω τρόπους και με όσους άλλους θα δούμε στη συνέχεια. Υπάρχει και η άλλη όψη, αυτή της εύρεσης της εξίσωσης διαφορών ενός δεδομένου διαγράμματος - το οποίο μπορεί να είναι από πολύ απλό ως αρκετά πολύπλοκο! Η διαδικασία που ακολουθούμε είναι όμως πάντα η ίδια:

- Τοποθετούμε στην έξοδο κάθε αθροιστή (πλην αυτού που παράγει την έξοδο) μια επιπλέον μεταβλητή.
- Γράφουμε τις εξισώσεις διαφορών που αναπαριστούν κάθε τέτοια μεταβλητή, μαζί με τη μεταβλητή της εξόδου.
- Μεταφέρουμε το σύνολο των εξισώσεων στο χώρο του Z. Οι εξισώσεις διαφορών μετατρέπονται σε απλές αλγεβρικές εξισώσεις.
- Προσπαθούμε να απαλείψουμε όλες τις ενδιάμεσες μεταβλητές που εισάγαμε και να καταλήξουμε σε μια εξίσωση που σχετίζει την είσοδο $X(z)$ με την έξοδο $Y(z)$.



Σχήμα 16.49: Direct Form I και II για το σύστημα του Παραδείγματος 16.32.

- Φέρνουμε την εξίσωση πίσω στο χώρο του χρόνου.

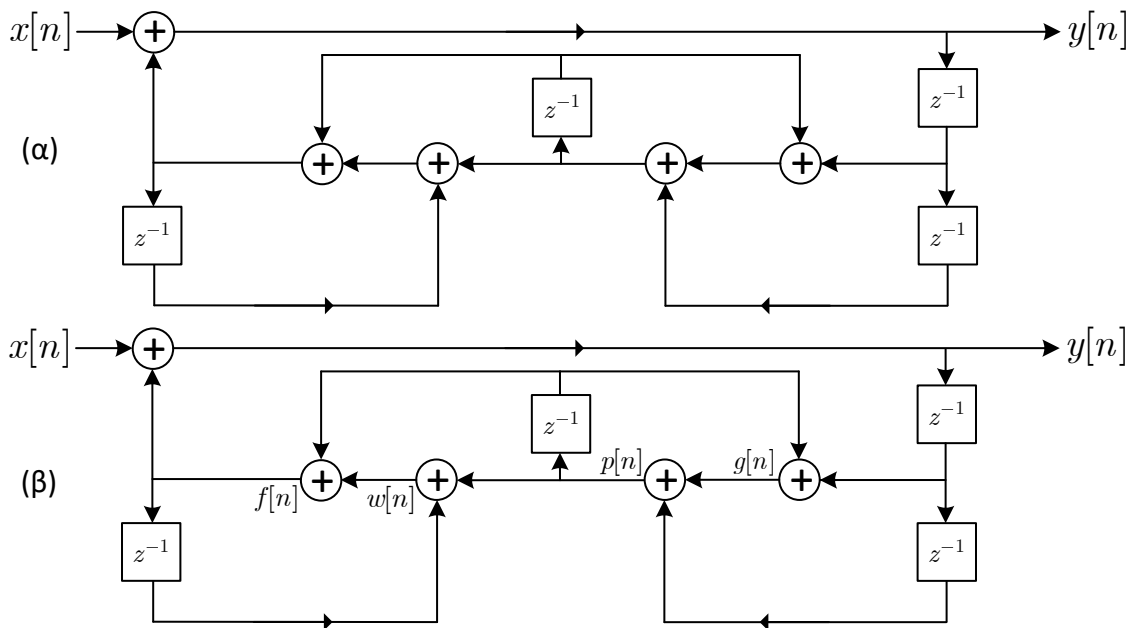
Ας δούμε ένα παράδειγμα.

Παράδειγμα 16.33:

Έστω το ΓΧΑ σύστημα που περιγράφεται από το Σχήμα 16.50(α). Βρείτε μια εξίσωση διαφορών που το περιγράφει.

Λύση:

Κατ' αρχάς αναγνωρίζουμε ότι το διάγραμμα αυτό δεν είναι μια από τις μορφές που έχουμε δει ως τώρα, οπότε δεν μπορούμε να βρούμε την εξίσωση διαφορών με απλή επισκόπηση. Βάζοντας επιπλέον μεταβλητές στο όπου χρειάζεται, καταλήγουμε στο Σχήμα 16.50(β).



Σχήμα 16.50: Διάγραμμα Παραδείγματος 16.33.

Οπότε θα είναι

$$y[n] = f[n] + x[n] \tag{16.441}$$

$$g[n] = y[n - 1] + p[n - 1] \tag{16.442}$$

$$p[n] = g[n] + y[n - 2] \tag{16.443}$$

$$w[n] = p[n] + f[n - 1] \quad (16.444)$$

$$f[n] = w[n] + p[n - 1] \quad (16.445)$$

στις οποίες κάνοντας μετασχ. Z, θα έχουμε

$$Y(z) = F(z) + X(z) \quad (16.446)$$

$$G(z) = z^{-1}Y(z) + z^{-1}P(z) \quad (16.447)$$

$$P(z) = G(z) + z^{-2}Y(z) \quad (16.448)$$

$$W(z) = P(z) + z^{-1}F(z) \quad (16.449)$$

$$F(z) = W(z) + z^{-1}P(z) \quad (16.450)$$

και αντικαθιστώντας, θα έχουμε

$$Y(z) = F(z) + X(z) \quad (16.451)$$

$$G(z) = z^{-1}Y(z) + z^{-1}(G(z) + z^{-2}Y(z)) = z^{-1}Y(z) + z^{-1}G(z) + z^{-3}Y(z) \quad (16.452)$$

$$= \frac{z^{-1} + z^{-3}}{1 - z^{-1}}Y(z) \quad (16.453)$$

$$F(z) = W(z) + z^{-1}P(z) = G(z) + z^{-2}Y(z) + z^{-1}F(z) + z^{-1}(G(z) + z^{-2}Y(z)) \quad (16.454)$$

$$= \frac{1}{1 - z^{-1}}(G(z) + z^{-2}Y(z) + z^{-1}G(z) + z^{-3}Y(z)) \quad (16.455)$$

$$= \frac{1}{1 - z^{-1}}\left(\frac{z^{-1} + z^{-3}}{1 - z^{-1}}Y(z) + z^{-2}Y(z) + z^{-1}\frac{z^{-1} + z^{-3}}{1 - z^{-1}}Y(z) + z^{-3}Y(z)\right) \quad (16.456)$$

$$= \frac{1}{1 - z^{-1}}\frac{z^{-1} + 2z^{-2} + z^{-3}}{1 - z^{-1}}Y(z) \quad (16.457)$$

και άρα τελικά από τη Σχέση (16.451), είναι

$$Y(z) = \frac{1}{1 - z^{-1}}\frac{z^{-1} + 2z^{-2} + z^{-3}}{1 - z^{-1}}Y(z) + X(z) \quad (16.458)$$

$$\frac{1 - 3z^{-1} - z^{-2} - z^{-3}}{1 - 2z^{-1} + z^{-2}}Y(z) = X(z) \quad (16.459)$$

$$(1 - 3z^{-1} - z^{-2} - z^{-3})Y(z) = (1 - 2z^{-1} + z^{-2})X(z) \quad (16.460)$$

Επιστρέφοντας στο πεδίο του χρόνου

$$y[n] - 3y[n - 1] - y[n - 2] - y[n - 3] = x[n] - 2x[n - 1] + x[n - 2] \quad (16.461)$$

που είναι και η ζητούμενη σχέση.

Όπως θα ξέρετε ήδη, τα συστήματα μπορούν να χωριστούν σε δυο κατηγορίες: τα Πεπερασμένης Κρουστικής Απόκρισης (FIR), και τα Άπειρης Κρουστικής Απόκρισης (IIR). Ας δούμε πόσο διαφορετικά είναι τα διαγράμματα για κάθε κατηγορία.

16.9.2 Βασικές Δομές IIR Συστημάτων

Ας ξεκινήσουμε από τα IIR συστήματα. Θα δούμε εδώ ότι υπάρχουν αρκετά διαφορετικά διαγράμματα που υλοποιούν μια συγκεκριμένη συνάρτηση μεταφοράς. Η κατάλληλη επιλογή διαγράμματος πρέπει να λάβει υπόψη της διάφορα θέματα, όπως η υπολογιστική πολυπλοκότητα. Για παράδειγμα, πολύ επιθυμητές είναι οι δομές με όσο γίνεται λιγότερους πολλαπλασιασμούς και καθυστερήσεις (θέσεις μνήμης). Αυτό ισχύει γιατί ο πολλαπλασιασμός είναι γενικά χρονοβόρος και κοστίζει σε ψηφιακό υλικό, και γιατί κάθε καθυστέρηση αντιστοιχεί σε έναν καταχωρητή μνήμης. Έτσι, η μείωση στον αριθμό των πολλαπλασιασμών συνεπάγεται αύξηση της ταχύτητας, και η μείωση στον αριθμό καθυστερήσεων συνεπάγεται μείωση στις απαιτήσεις μνήμης.

Οι προδιαγραφές που μπορεί κανείς να θέσει σε μια δομή δεν εξαντλούνται στις παραπάνω, αν και αυτές θα μας απασχολήσουν στο παρόν σύγγραμμα. Παραδείγματος χάριν, σε μοντέρνες υλοποιήσεις υλικού το εμβαδό του chip είναι σημαντική παράμετρος, ενώ η απλότητα και η αριθρωτή υλοποίηση είναι επίσης επιθυμητά στοιχεία. Ο αναγνώστης που ενδιαφέρεται για τέτοιες λεπτομέρειες μπορεί να δει τα [104], [103], [80], [108].

16.9.2.1 Direct Forms

Ήδη έχουμε συζητήσει για τη μορφή ενός Direct Form διαγράμματος, τόσο του Direct Form I όσο και του II. Έστω η εξίσωση διαφορών

$$y[n] - \sum_{k=1}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k] \quad (16.462)$$

η οποία ανταποκρίνεται στη συνάρτηση μεταφοράς

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 - \sum_{k=1}^N a_k z^{-k}} \quad (16.463)$$

Η σχεδίαση των διαγραμμάτων είναι τέτοια ώστε κάθε αθροιστής να έχει το πολύ δυο εισόδους - στην πράξη μπορεί να έχει περισσότερες εισόδους, αλλά όπως είπαμε και νωρίτερα, αυτή η “σύμβαση” των δυο εισόδων έχει ως αποτέλεσμα διαγράμματα που είναι πιο κοντά σε προγράμματα και αρχιτεκτονικές υλοποίησης των υπολογισμών που απαιτούνται για μια εξίσωση διαφορών που αναπαριστά ένα διάγραμμα. Ας δούμε ένα παράδειγμα.

Παράδειγμα 16.34:

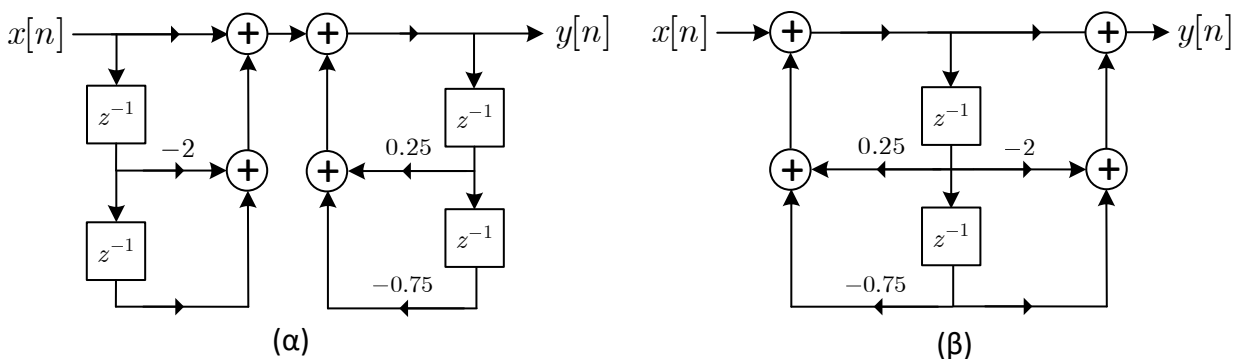
Έστω η συνάρτηση μεταφοράς

$$H(z) = \frac{1 - 2z^{-1} + z^{-2}}{1 - 0.25z^{-1} + 0.75z^{-2}} \quad (16.464)$$

Σχεδιάστε τα διαγράμματα Direct Form I, II.

Λύση:

Μπορούμε να σχεδιάσουμε τα διαγράμματα Direct Form I, II απευθείας, καθώς η συνάρτηση μεταφοράς είναι γραμμένη σε βολική μορφή. Έτσι, καταλήγουμε στο Σχήμα 16.51.



Σχήμα 16.51: Direct Form I, II του Παραδείγματος 16.34.

16.9.2.2 Μορφή σε Σειρά - Cascade Form

Τα διαγράμματα Direct Form προέρχονται από την απλή υλοποίηση των πολωνύμων αριθμητή και παρονομαστή, όπως αυτοί παρουσιάζονται στη συνάρτηση μεταφοράς. Αν όμως παραγοντοποιήσουμε αριθμητή και παρονομαστή, τότε θα έχουμε

$$H(z) = A \frac{\prod_{k=1}^{M_1} (1 - e_k z^{-1}) \prod_{k=1}^{M_2} (1 - g_k z^{-1})(1 - g_k^* z^{-1})}{\prod_{k=1}^{N_1} (1 - c_k z^{-1}) \prod_{k=1}^{N_2} (1 - d_k z^{-1})(1 - d_k^* z^{-1})} \quad (16.465)$$

με $M = M_1 + 2M_2$ και $N = N_1 + 2N_2$. Σε αυτήν την έκφραση, οι όροι πρώτης τάξης αναπαριστούν μηδενικά, e_k , και πόλους c_k , στον πραγματικό άξονα. Οι όροι δεύτερης τάξης αναπαριστούν συζυγή ζεύγη μιγαδικών μηδενικών, g_k, g_k^* , και πόλων d_k, d_k^* . Προφανώς η παραπάνω συνάρτηση μεταφοράς ανταποκρίνεται σε ένα πραγματικό

σύστημα. Η γραφή αυτή αναπαριστά μια κατηγορία δομών που περιέχουν σε σειρά όρους πρώτης και δεύτερης τάξης, ως υποσυστήματα. Υπάρχει αρκετή ελευθερία στην επιλογή της σύνθεσης των υποσυστημάτων και στη σειρά με την οποία υλοποιούνται. Στην πράξη βέβαια είναι συχνά επιθυμητή η υλοποίηση σε σειρά με τον ελάχιστο δυνατό αριθμό υπολογισμών και μνήμης. Μια δομή σε σειρά - που έχει πολλά πλεονεκτήματα - μπορεί να πραγματοποιηθεί συνδυάζοντας ζεύγη πραγματικών όρων και συζυγών μιγαδικών σε όρους δεύτερης τάξης, έτσι ώστε η Σχέση (16.465) να γράφει ως

$$H(z) = \prod_{k=1}^{N_s} \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 - a_{1k}z^{-1} - a_{2k}z^{-2}} \quad (16.466)$$

όπου $N_s = \lfloor (N+1)/2 \rfloor$ είναι ο μεγαλύτερος ακέραιος. Γράφοντας το $H(z)$ στη μορφή αυτή, υποθέσαμε ότι $M \leq N$ και ότι οι πραγματικοί πόλοι και τα πραγματικά μηδενικά συνδυάζονται σε ζεύγη. Τα επιμέρους δεύτερης τάξης υποσυστήματα μπορούν να υλοποιηθούν είτε με Direct Form I είτε με Direct Form II, με τη δεύτερη να προτιμάται όταν θέλουμε να είμαστε οικονομικοί σε στοιχεία καθυστέρησης και στοιχεία πολλαπλασιασμού. Ας δούμε ένα παράδειγμα.

Παράδειγμα 16.35:

Έστω το σύστημα

$$H(z) = \frac{1 - 0.5z^{-1} - 0.5z^{-2}}{1 + 0.75z^{-1} - 0.25z^{-2}} \quad (16.467)$$

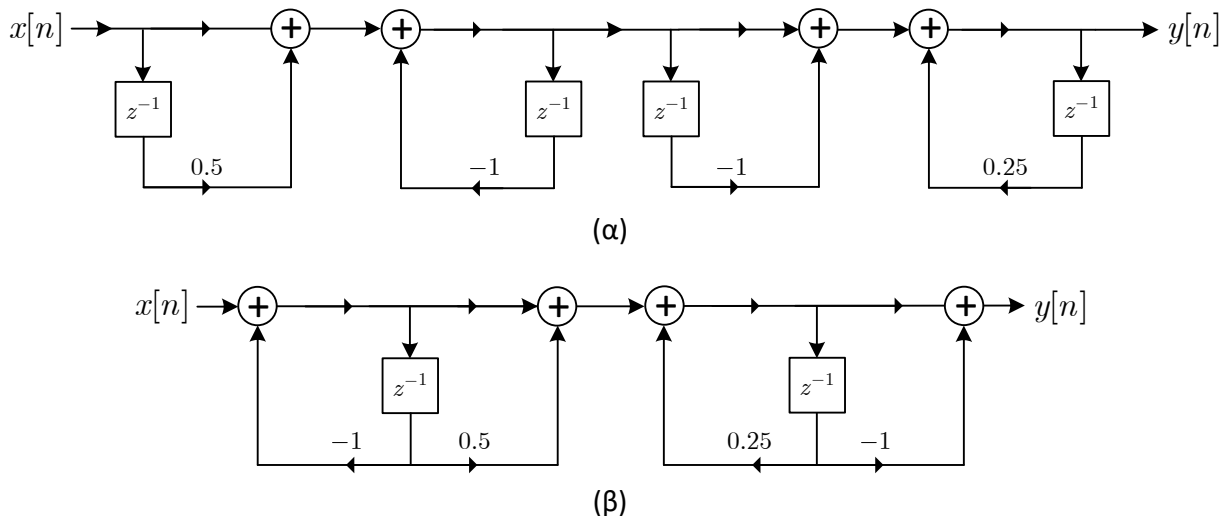
Υλοποιείστε το παραπάνω σύστημα σε σειρά με χρήση Direct Form I, II διαγραμμάτων.

Λύση:

Το παραπάνω σύστημα μπορεί να γράφει ως

$$H(z) = \frac{(1 + 0.5z^{-1})(1 - z^{-1})}{(1 + z^{-1})(1 - 0.25z^{-1})} \quad (16.468)$$

Η υλοποίηση σε σειρά με χρήση Direct Form I, II διαγράμματα φαίνεται στο Σχήμα 16.52.



Σχήμα 16.52: Διάγραμμα σε σειρά με υποσυστήματα Direct Form I, II Παραδείγματος 16.35.

Ένας άλλος ορισμός της μορφής σε σειρά είναι ως ακολούθως

$$H(z) = b_0 \prod_{k=1}^{N_s} \frac{1 + \hat{b}_{1k}z^{-1} + \hat{b}_{2k}z^{-2}}{1 - a_{1k}z^{-1} - a_{2k}z^{-2}} \quad (16.469)$$

όπου έχουμε διαρέσει κάθε όρο του αριθμητή με το b_{0k} , παίρνοντας έτσι νέους συντελεστές $\hat{b}_{ik} = \frac{b_{ik}}{b_{0k}}$, για $i = 1, 2$ και $k = 1, 2, \dots, N_s$. Αυτή η μορφή του $H(z)$ προτείνει μια σχεδίαση σε σειρά με υποσυστήματα δεύτερης τάξης που απαιτούν τέσσερις πολλαπλασιασμούς, με συνολική σταθερά b_0 για όλο το σύστημα. Αυτή η μορφή σε σειρά έχει τον ίδιο αριθμό πολλαπλασιασμών με τις Direct Form δομές.

16.9.2.3 Παράλληλη Μορφή - Parallel Form

Στη συζήτησή μας περί ανάλυσης συστημάτων στο χώρο του Z , χρησιμοποιούσαμε συχνά το ανάπτυγμα σε μερικά κλάσματα ως αναπαράσταση πολύπλοκων συναρτήσεων μεταφοράς. Ας δούμε αν αυτό το ανάπτυγμα μας προτείνει κάποια δομή σχεδίασης που είναι χρήσιμη. Ας θυμηθούμε ότι η Σχέση (16.463) μπορεί να γραφεί ως

$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_1} \frac{A_k}{1 - c_k z^{-1}} + \sum_{k=1}^{N_2} \frac{B_k(1 - e_k z^{-1})}{(1 - d_k z^{-1})(1 - d_k^* z^{-1})} \quad (16.470)$$

με $N = N_1 + 2N_2$. Αν $M \geq N$, τότε $N_p = M - N$, και άρα ο πρώτος όρος υπάρχει, ειδικά $N_p < 0$ και τότε τον απορρίπτουμε από την παραπάνω σχέση. Αν οι συντελεστές a_k, b_k της Σχέσης (16.463) είναι πραγματικοί αριθμοί, τότε οι συντελεστές A_k, B_k, C_k, c_k, e_k του αναπτύγματος είναι πραγματικοί αριθμοί. Σε αυτήν την περίπτωση, η συνάρτηση μεταφοράς μπορεί να ερμηνευθεί ως μια αναπαράσταση ενός παράλληλου συνδυασμού IIR υποσυστημάτων πρώτης και δεύτερης τάξης, με την πιθανότητα να υπάρχουν επιπλέον N_p κλάδοι καθυστέρησης της εισόδου. Εναλλακτικά, μπορούμε να ομαδοποιήσουμε τους πραγματικούς πόλους σε ζεύγη, έτσι ώστε το $H(z)$ να γραφεί ως

$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_s} \frac{e_{0k} + e_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}} \quad (16.471)$$

όπου $N_s = \lfloor (N + 1)/2 \rfloor$. Ξανά, αν $N_p = M - N < 0$, ο πρώτος όρος δεν περιλαμβάνεται στην παραπάνω σχέση.

Ας δούμε ένα παράδειγμα.

Παράδειγμα 16.36:

Έστω το σύστημα

$$H(z) = \frac{1 + 2.5z^{-1} - z^{-2}}{1 - 0.25z^{-2}} \quad (16.472)$$

Υλοποιήστε το σε παράλληλη μορφή.

Λύση:

Το ανάπτυγμα σε μερικά κλάσματα μας δίνει

$$H(z) = 4 + \frac{-3 + 2.5z^{-1}}{1 - 0.25z^{-2}} \quad (16.473)$$

και μπορεί να υλοποιηθεί ως Parallel Form δομή όπως στο Σχήμα 16.53. Επίσης, το ίδιο σύστημα μπορεί να γραφεί ως άθροισμα στοιχείων πρώτης τάξης, δηλ. ως

$$H(z) = 4 + \frac{1}{1 - 0.5z^{-1}} - \frac{4}{1 + 0.5z^{-1}} \quad (16.474)$$

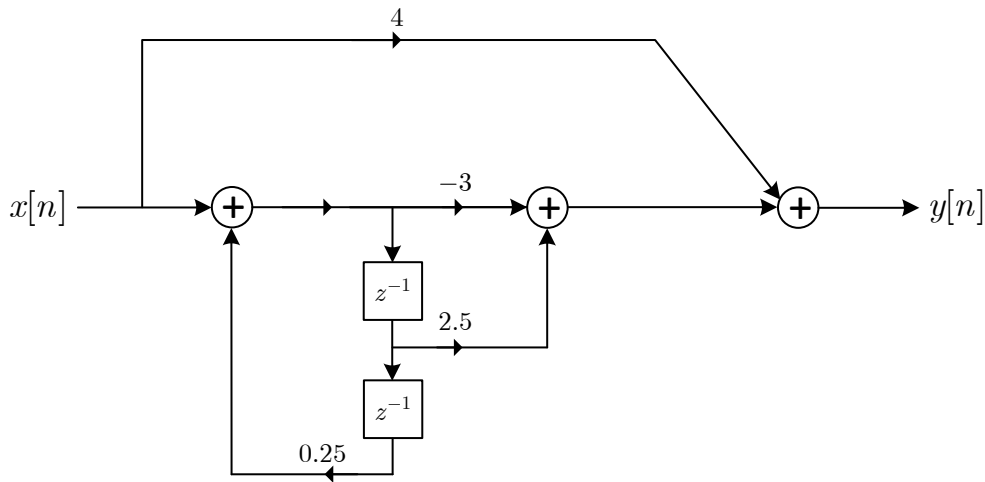
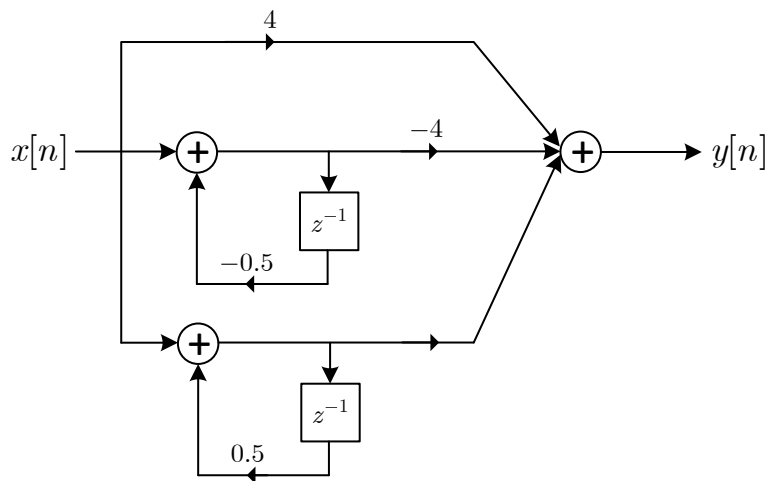
το οποίο μπορεί να υλοποιηθεί ως Parallel Form διάγραμμα όπως φαίνεται στο Σχήμα 16.54.

16.9.2.4 Ανάστροφη Μορφή - Transposed Form

Έχουμε δει ως τώρα πόσες διαφορετικές υλοποιήσεις μιας δεδομένης εξίσωσης διαφορών υπάρχουν. Στην πραγματικότητα υπάρχουν ακόμα περισσότερες τις οποίες δεν μπορούμε να εξαντλήσουμε εδώ. Όμως ας αναφέρουμε μια ακόμα τέτοια διαδικασία, που ονομάζεται *αναστροφή - reversal*, και μας παρέχει μερικές καλές εναλλακτικές μορφές διαγραμμάτων, με ενδιαφέρουσες ιδιότητες. Ας τις δούμε.

Οι κανόνες για την εύρεση του αναστροφικού διαγράμματος (οι οποίοι δεν αποδεικνύονται εδώ, μια και προέρχονται από τη θεωρία γράφων) είναι οι εξής:

1. Αναστρέφουμε τη φορά όλων των κλάδων του διαγράμματος.

Σχήμα 16.53: *Parallel Form I* διάγραμμα Παραδείγματος 16.36.Σχήμα 16.54: *Parallel Form II* διάγραμμα Παραδείγματος 16.36.

2. Κάθε αθροιστής μετατρέπεται σε διακλαδωτής, και το αντίστροφο.
3. Αλλάζουμε θέση μεταξύ εισόδου και εξόδου.
4. Σχεδιάζουμε το διάγραμμα ξανά, αναστρέφοντας το ώστε να παρουσιάζεται η είσοδος στα αριστερά.

Ας το δούμε στην πράξη σε ένα παράδειγμα.

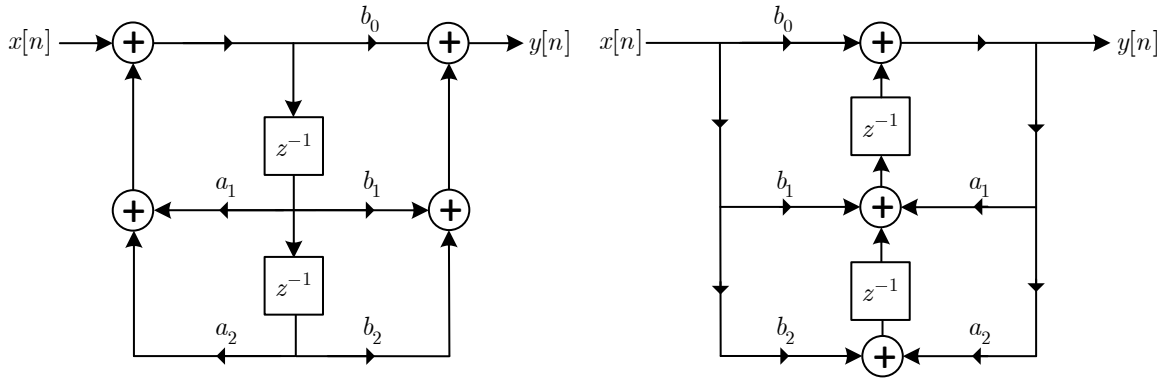
Παράδειγμα 16.37:

Έστω το Direct Form II διάγραμμα του Σχήματος 16.55(α), που αντιστοιχεί σε ένα σύστημα δευτέρας τάξης με πόλους και μηδενικά. Σχεδιάστε την ανάστροφη μορφή του διαγράμματός του.

Λύση:

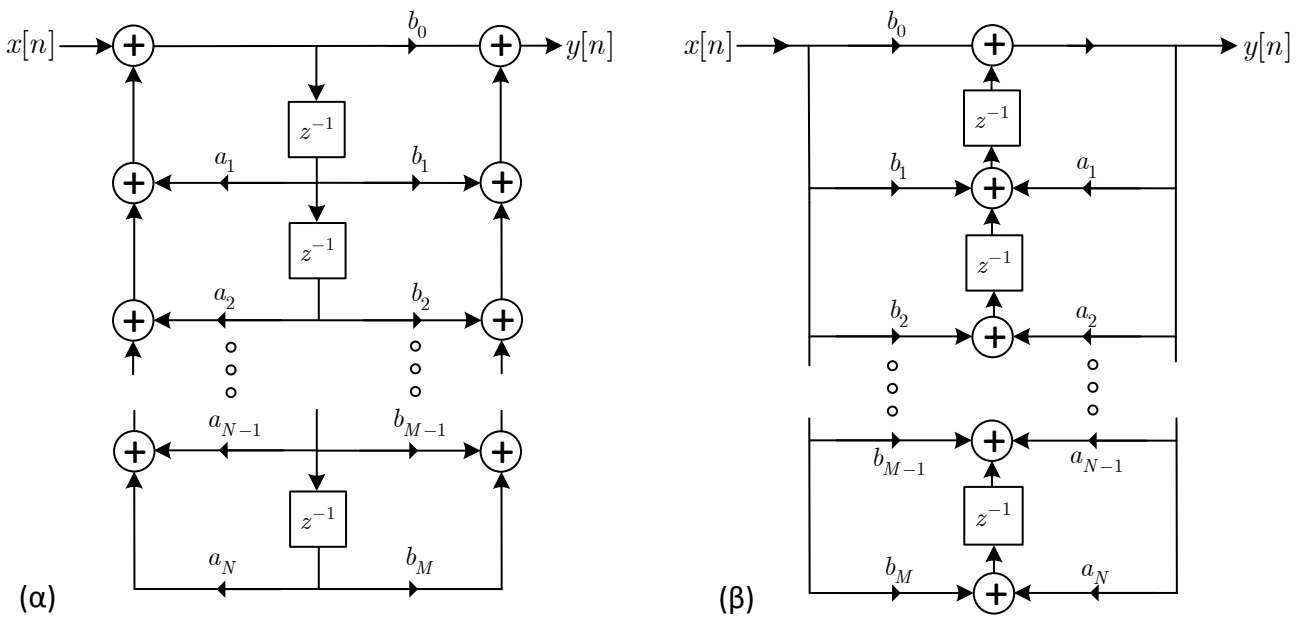
Εφαρμόζοντας ξανά τους κανόνες 1–4, καταλήγουμε στο ανάστροφο διάγραμμα του Σχήματος 16.55(β). Μπορείτε να δείξετε ότι οι εξισώσεις διαφορών των δυο διαγραμμάτων περιγράφουν το ίδιο σύστημα;

Η εφαρμογή των κανόνων για την εύρεση του αναστρώφου διαγράμματος μπορεί να γίνει σε οποιαδήποτε από τα διαγράμματα που συναντήσαμε ως τώρα. Ας γενικεύσουμε αυτήν την εφαρμογή, ξεκινώντας από ένα Direct



Σχήμα 16.55: (α) Direct Form II και (β) ανάστροφο διάγραμμα Παραδείγματος 16.37.

Form II διάγραμμα, και καταλήγοντας σε μια transposed δομή. Το Σχήμα 16.56 δείχνει τη μετατροπή αυτή σε ένα γενικότερο πλαίσιο. Όμοια συμβαίνει και η μετατροπή ενός Direct Form I διαγράμματος σε transposed μορφή.



Σχήμα 16.56: (α) Direct Form II και (β) ανάστροφο διάγραμμα.

Μπορούμε να κάνουμε μια ενδιαφέρουσα παρατήρηση αν συγκρίνουμε τα δυο διαγράμματα του Σχήματος 16.56. Ενώ στο Direct Form II διάγραμμα, υλοποιούμε πρώτα τους πόλους και μετά τα μηδενικά, βλέπουμε ότι στο transposed διάγραμμα υλοποιούμε πρώτα τα μηδενικά και μετά τους πόλους, χωρίς την επιπλέον χρήση μνήμης που είχαμε όταν το κάναμε αυτό στη Direct Form I σχεδίαση! Αυτό είναι μερικές φορές σημαντικό όταν υλοποιούμε αυτά τα συστήματα με αναλογικά ή ψηφιακά κυκλώματα.

Τέλος, η αναστροφή διαγράμματος τονίζει ακόμα περισσότερο τη μεγάλη ποικιλία των δομών υλοποίησης που υπάρχει για ένα δεδομένο σύστημα που περιγράφεται από μια ρητή συνάρτηση μεταφοράς. Η αναστροφή μας παρέχει μια απλή διαδικασία για τη δημιουργία νέων δομών συστημάτων. Τα προβλήματα που προέκυψαν από την υλοποίηση τέτοιων συστημάτων σε μηχανές πεπερασμένης ακρίβειας έδωσαν κίνητρο στους ερευνητές να αναπτύξουν ακόμα περισσότερες κατηγορίες δομών, που δεν είναι βολικό να συζητήσουμε εδώ, καθώς μας αρκεί να γνωρίζουμε τις πιο συχνά χρησιμοποιούμενες δομές συστημάτων.

16.9.3 Βασικές Δομές FIR Συστημάτων

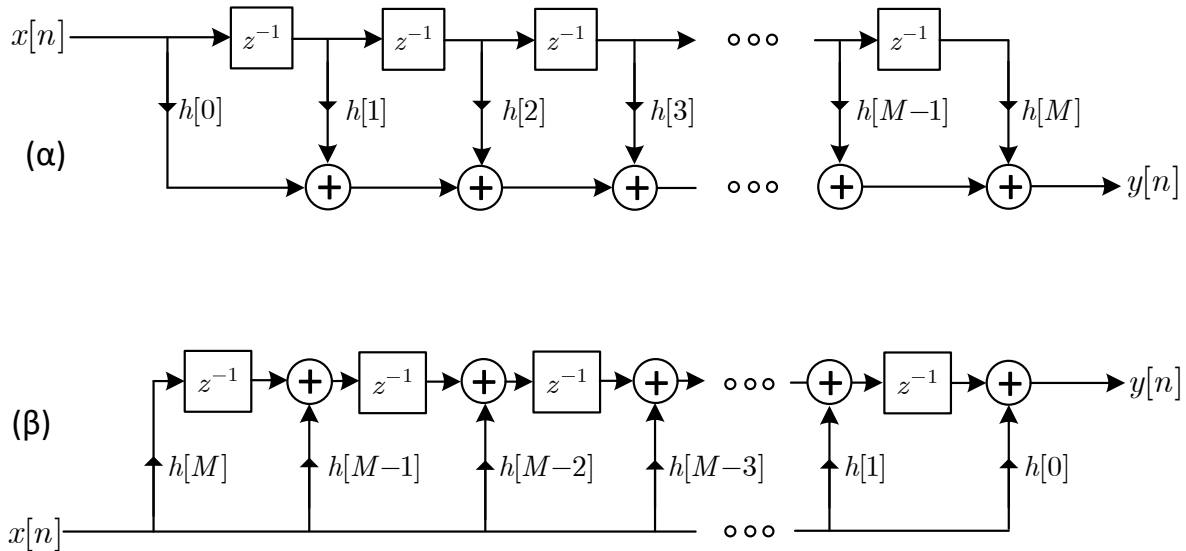
Για τα FIR συστήματα, τα πράγματα είναι σχετικά απλούστερα. Παρ'ότι οι δομές Direct Form, σε σειρά, και παράλληλα που συζητήσαμε προηγουμένως αφορούν IIR συστήματα, τα οποία φυσικά περιέχουν τα FIR ως ειδική περίπτωση, υπάρχουν ανεξάρτητες δομές, ειδικές για FIR συστήματα.

16.9.3.1 Direct Forms

Για αιτιατά FIR συστήματα, η συνάρτηση μεταφοράς έχει μόνο μηδενικά, και πόλους μόνο στο $z = 0$. Οπότε η εξίσωση διαφορών γράφεται ως

$$y[n] = \sum_{k=0}^M b_k x[n - k] \tag{16.475}$$

Σε αυτήν την περίπτωση, οι μορφές Direct Form I, II που έχουμε ήδη δει, μετατρέπονται σε μια μόνο Direct Form, αυτή του Σχήματος 16.57, η οποία παρουσιάζεται μαζί με την ανάστροφη μορφή της, ακολουθώντας τους κανόνες που περιγράψαμε νωρίτερα. Η Direct Form σχεδίαση λέγεται και Transversal Form.



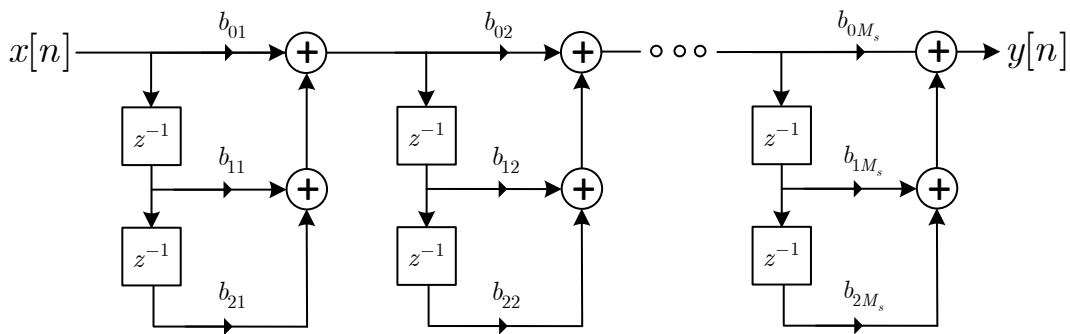
Σχήμα 16.57: (α) Direct Form και (β) ανάστροφο διάγραμμα για ένα FIR σύστημα.

16.9.3.2 Μορφή σε Σειρά - Cascade Form

Η μορφή σε σειρά ενός FIR συστήματος προέρχεται από την παραγοντοποίηση του πολυωνύμου της συνάρτησης μεταφοράς, δηλ.

$$H(z) = \sum_{n=0}^M h[n]z^{-n} = \prod_{k=1}^{M_s} (b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}) \tag{16.476}$$

όπου $M_s = \lfloor (M + 1)/2 \rfloor$. Μια γενική αναπαράσταση της μορφής σε σειρά φαίνεται στο Σχήμα 16.58.



Σχήμα 16.58: Μορφή σε σειρά για ένα FIR σύστημα.

16.9.3.3 Δομές για Συστήματα Γραμμικής Φάσης

Μια ειδική κατηγορία FIR συστημάτων είναι αυτά που έχουν γραμμική φάση. Έχουμε δείξει ότι τα συστήματα αυτά ικανοποιούν κάποια συμμετρία, όπως

$$h[M - n] = h[n], \quad n = 0, 1, \dots, M \tag{16.477}$$

για τα Τύπου I, II, ή

$$h[M - n] = -h[n], \quad n = 0, 1, \dots, M \tag{16.478}$$

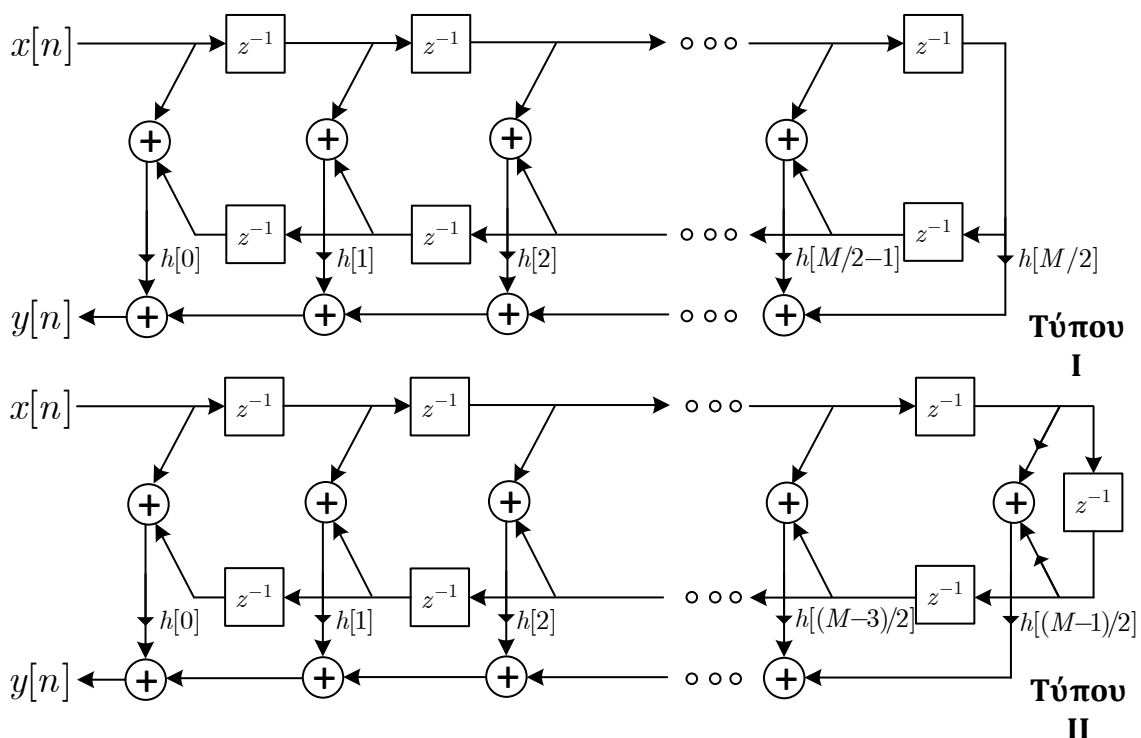
για τα Τύπου III, IV. Οποιαδήποτε κι αν ισχύει από τις δυο συμμετρίες, ο αριθμός των πολλαπλασιασμών που απαιτούνται για την υλοποίησή τους μπορεί να μειωθεί σχεδόν στο μισό! Για να το δούμε αυτό, ας θεωρήσουμε ότι το M είναι άρτιος ακέραιος, και αντιστοιχεί σε ένα Τύπου I ή III σύστημα. Τότε:

$$y[n] = \sum_{k=0}^M h[k]x[n - k] \tag{16.479}$$

$$= \sum_{k=0}^{M/2-1} h[k]x[n - k] + h[M/2]x[n - M/2] + \sum_{k=M/2+1}^M h[k]x[n - k] \tag{16.480}$$

$$= \sum_{k=0}^{M/2-1} h[k]x[n - k] + h[M/2]x[n - M/2] + \sum_{k=0}^{M/2-1} h[M - k]x[n - M + k] \tag{16.481}$$

Για Τύπου I συστήματα, έχουμε



Σχήμα 16.59: Direct Forms για ένα FIR σύστημα γραμμικής φάσης Τύπου I και II.

$$y[n] = \sum_{k=0}^{M/2-1} h[k](x[n - k] + x[n - M + k]) + h[M/2]x[n - M/2] \tag{16.482}$$

ενώ για Τύπου III συστήματα, έχουμε

$$y[n] = \sum_{k=0}^{M/2-1} h[k](x[n - k] + x[n - M + k]) \tag{16.483}$$

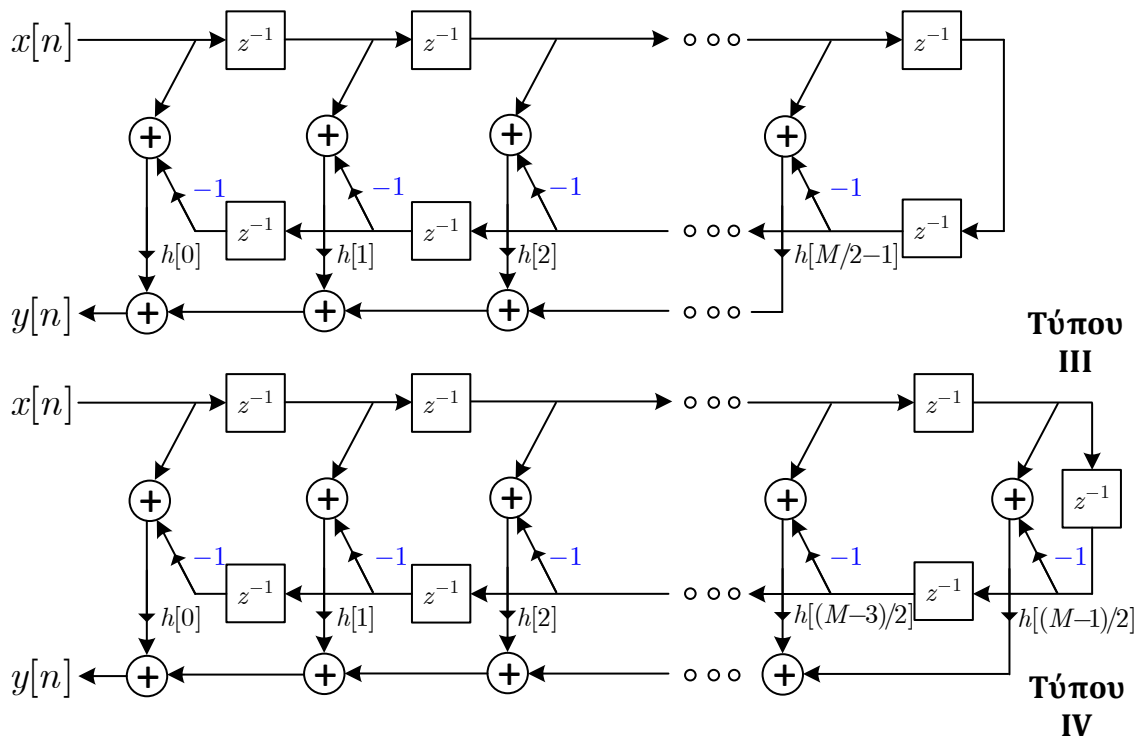
Αν δουλέψουμε αντίστοιχα για τα Τύπου II, IV συστήματα (με M περιττό δηλαδή), θα καταλήξουμε ότι για τα Τύπου II συστήματα, έχουμε

$$y[n] = \sum_{k=0}^{(M-1)/2} h[k](x[n-k] + x[n-M+k]) \quad (16.484)$$

ενώ για τα Τύπου IV, θα είναι

$$y[n] = \sum_{k=0}^{(M-1)/2} h[k](x[n-k] - x[n-M+k]) \quad (16.485)$$

Μπορείτε να αποδείξετε τις παραπάνω σχέσεις στην Άσκηση 16.41. Προσέξτε ότι τα παραπάνω συστήματα έχουν $M/2 + 1, M/2$ ή $(M + 1)/2$ πολλαπλασιασμούς, αντί για M πολλαπλασιασμούς που θα είχαν αν τα υλοποιούσαμε σε γενική Direct Form. Τα διαγράμματα των παραπάνω συστημάτων Τύπου I και II φαίνονται στο Σχήμα 16.59, ενώ για τα αντίστοιχα των Τύπων III και IV δείτε το Σχήμα 16.60



Σχήμα 16.60: Direct Forms για ένα FIR σύστημα γραμμικής φάσης Τύπου III και IV.

Ας δούμε ένα παράδειγμα.

Παράδειγμα 16.38:

Έστω το ΓΧΑ σύστημα που δίνεται από την κρουστική απόκριση

$$h[n] = \delta[n] + 2\delta[n-1] + 2\delta[n-3] + \delta[n-4] \quad (16.486)$$

Συζητήστε και σχεδιάστε τη βέλτιστη (από πλευράς οικονομίας πόρων) υλοποίησή του.

Λύση:

Είναι εμφανές ότι το σύστημα είναι FIR και παρουσιάζει συμμετρία γύρω από το δείγμα $n_0 = 2$. Η παράμετρος M ισούται με 4, άρα είναι ένα σύστημα γραμμικής φάσης Τύπου I. Μια κλασική υλοποίηση Direct Form θα απαιτούσε:

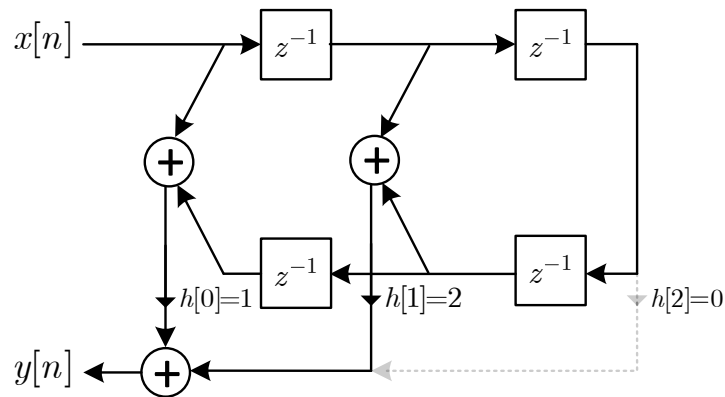
- 2 πολλαπλασιασμούς (εξαιρώντας πολλαπλασιασμούς με τη μονάδα)
- 4 θέσεις μνήμης

- 3 αθροιστές

Αν εκμεταλλευτούμε την υλοποίηση που συζητήσαμε σε αυτήν την Παράγραφο, τότε μπορούμε να εξοικονομήσουμε πόρους. Η εξίσωση διαφορών που περιγράφει το σύστημα αυτό είναι η

$$y[n] = x[n] + 2x[n-1] + 2x[n-3] + x[n-4] = h[0]x[n] + h[1]x[n-1] + h[3]x[n-3] + h[4]x[n-4] \quad (16.487)$$

Το διάγραμμα φαίνεται στο Σχήμα 16.61. Παρατηρήστε ότι λόγω του ότι $h[M/2] = h[2] = 0$, ένας από τους



Σχήμα 16.61: FIR σύστημα γραμμικής φάσης Τύπου I Παραδείγματος 16.38.

κλάδους (σημειωμένος με γκρίζο χρώμα) του διαγράμματος μπορεί να παραλειφθεί. Το διάγραμμα αυτό διαθέτει

- 4 θέσεις μνήμης
- 1 πολλαπλασιαστή
- 3 αθροιστές

Έτσι εξοικονομούμε μια πράξη πολλαπλασιασμού σε σχέση με μια Direct Form υλοποίηση!