

# Ψηφιακή Επεξεργασία Σήματος & Εικόνας

## Μάθημα 21<sup>ο</sup>

Ελληνικό Μεσογειακό Πανεπιστήμιο  
Τμήμα Ηλεκτρονικών Μηχανικών

Δομές Πραγματοποίησης Ψηφιακών Φίλτρων

A. Μπακλέζος

[abaklezos@hmu.gr](mailto:abaklezos@hmu.gr)

## Δομές Πραγματοποίησης Ψηφιακών Φίλτρων

- Οποιοδήποτε αιτιατό γραμμικό χρονικά αμετάβλητο σύστημα διακριτού χρόνου χαρακτηρίζεται πλήρως από τη συνάρτηση μεταφοράς του η οποία έχει τη γενική μορφή:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}$$

Βέβαια, τα πολυώνυμα αριθμητή και παρονομαστή μπορούν να παραγοντοποιηθούν ως προς τις ρίζες τους και έτσι η συνάρτηση μεταφοράς να γραφεί στη μορφή:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 (1 - \rho_1 z^{-1})(1 - \rho_2 z^{-1}) \dots (1 - \rho_M z^{-1})}{(1 - r_0 z^{-1})(1 - r_1 z^{-1})(1 - r_2 z^{-1}) \dots (1 - r_N z^{-1})}$$

Επιπλέον είναι εφικτή και η ακόλουθη διάσπαση σε άθροισμα απλούστερων κλασμάτων:

$$H(z) = \frac{B(z)}{A(z)} = \frac{A_0}{(1 - r_0 z^{-1})} + \frac{A_1}{(1 - r_1 z^{-1})} + \dots + \frac{A_N}{(1 - r_N z^{-1})}$$

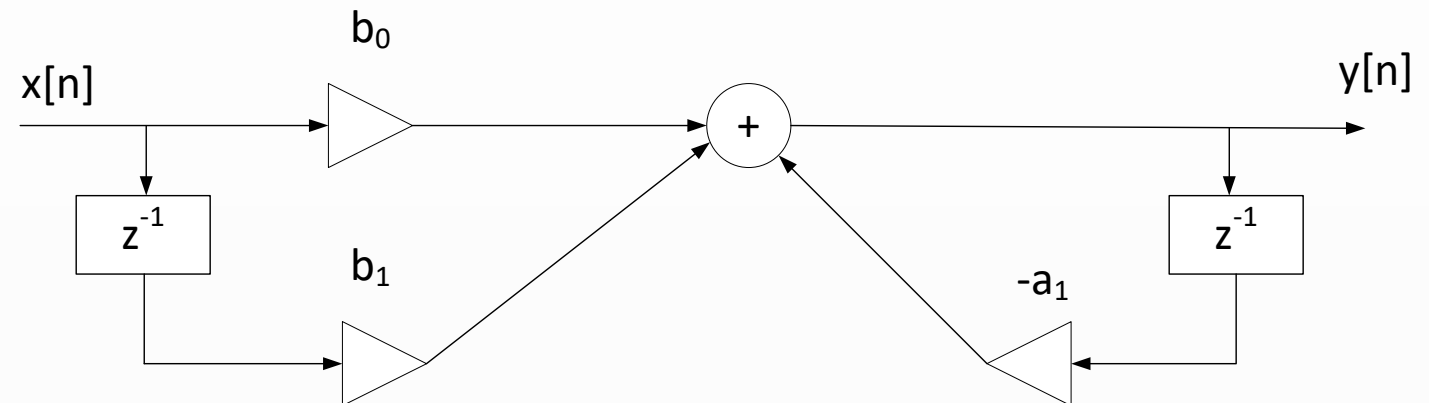
Οι παραπάνω μορφές αποτελούν μερικά παραδείγματα από τα οποία μπορούμε να οδηγηθούμε σε διαφορετικές πραγματοποιήσεις του συστήματος στο οποίο αντιστοιχούν.

## Δομές Πραγματοποίησης Ψηφιακών Φίλτρων

- Η δομή (δηλαδή το μπλοκ διάγραμμα) πρώτο βήμα προς την υλοποίηση του φίλτρου
- Είναι σχηματικά διαγράμματα για τους διαφορετικούς αλλά ισοδύναμους τρόπους που μπορεί να οργανωθεί η συνάρτηση μεταφοράς  $H(z)$ .

Π.χ. το φίλτρο IIR

$$y[n] = -a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$



Γνωρίζοντας την αρχική συνθήκη  $y[-1]$  και την είσοδο  $x[n]$ ,  $n \geq -1$  υπολογίζεται η έξοδος  $\forall n \geq 0$

$$y[0] = -a_1 y[-1] + b_0 x[0] + b_1 x[-1]$$

$$y[1] = -a_1 y[0] + b_0 x[1] + b_1 x[0]$$

$$y[2] = -a_1 y[1] + b_0 x[2] + b_1 x[1]$$

...

## Δομές Πραγματοποίησης Ψηφιακών Φίλτρων

➤ Γιατί δομές ;

Ευκολία :

- απευθείας καταγραφή του αλγορίθμου που πρέπει να υλοποιηθεί με την επισκόπηση του διαγράμματος
- στην ανάλυση και συσχέτιση εισόδου – εξόδου
- την επισκόπηση και στις μεταβολές στη δομή ώστε να υλοποιηθεί διαφορετικός αλγόριθμος
- στον προσδιορισμό των απαιτήσεων σε υλικό
- σύγκριση μεταξύ ισοδυνάμων υλοποιήσεων

## Δομές Πραγματοποίησης Ψηφιακών Φίλτρων

- Οι διαφορετικές υλοποιήσεις που προκύπτουν κάθε φορά διαφέρουν ως προς ένα σύνολο χαρακτηριστικών μεταξύ των οποίων :
  - η υπολογιστική πολυπλοκότητα (πλήθος προσθέσεων και πολλαπλασιασμών)
  - το πλήθος των στοιχείων καθυστέρησης (μνήμη) που χρησιμοποιούν
  - η ευαισθησία της υλοποίησης ως προς σφάλματα κβαντισμού
- ο βαθμός παραλληλοποίησης ο οποίος επιτυγχάνεται
- ο βαθμός pipelining που επιτυγχάνεται

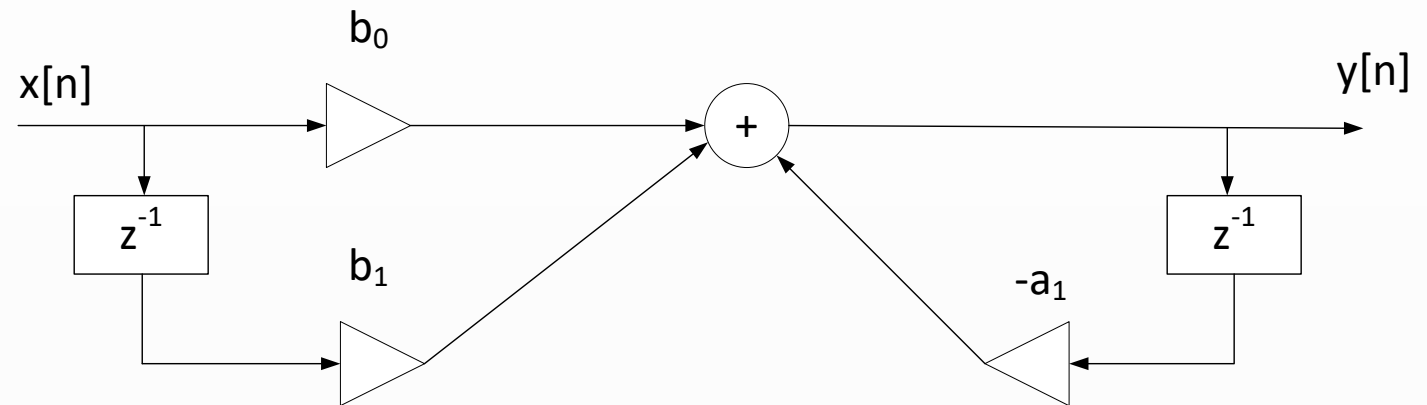
## Δομές Πραγματοποίησης Ψηφιακών Φίλτρων

- Υπάρχουν άπειρο πλήθος ισοδύναμων δομών που πραγματοποιούν την ίδια συνάρτηση μεταφοράς
- Υπό την προϋπόθεση της άπειρης ακρίβειας όλες οι υλοποιήσεις θα είχαν την ίδια συμπεριφορά
- Όμως στη πράξη λόγω των περιορισμών του finite word-length οι δομές συμπεριφέρονται διαφορετικά
- Έτσι έχει σημασία η κατάλληλη επιλογή της δομής

## Δομές Πραγματοποίησης Ψηφιακών Φίλτρων

Κανονικές δομές : οι δομές που το πλήθος των καθυστερήσεων (delays) στο block διάγραμμα είναι ίσες με τη τάξη της συνάρτησης μεταφοράς

$$y[n] = -a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$



Μη κανονική δομή γιατί χρησιμοποιεί δύο στοιχεία μνήμης για φίλτρο τάξης 1.

## Δομές Πραγματοποίησης Ψηφιακών Φίλτρων

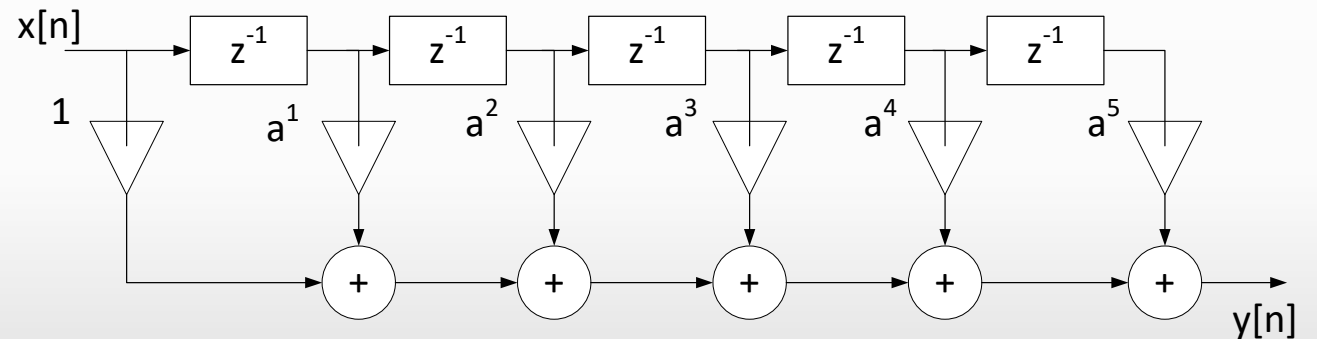
Κανονικές δομές : οι δομές που το πλήθος των καθυστερήσεων (delays) στο block διάγραμμα είναι ίσες με τη τάξη της συνάρτησης μεταφοράς

$$h[n] = \begin{cases} a^n, & 0 \leq n \leq 5 \\ 0, & \text{αλλού} \end{cases}$$

Η κρουστική απόκριση γράφεται:

$$h[n] = a^n [u[n] - n[n - 6]] = \delta(0) + a\delta(1) + a^2\delta(2) + a^3\delta(3) + a^4\delta(4) + a^5\delta(5)$$

αποκλειστικά στα FIR φίλτρα :  
Η Direct Form I είναι κανονική δομή





## Δομές Πραγματοποίησης FIR φίλτρων

Για τα FIR συστήματα οι σημαντικότερες μορφές είναι οι :

**Άμεση Μορφή I (Direct Form I):** Προκύπτει από την άμεση εφαρμογή της γραμμικής εξίσωσης διαφορών με σταθερούς συντελεστές σε διάγραμμα βαθμίδων.

**Γραμμικής φάσης:** Όταν η κρουστική απόκριση ικανοποιεί κάποιες ιδιότητες συμμετρίας τότε η φάση του συστήματος είναι γραμμική. Λόγω της συμμετρίας μειώνεται το πλήθος των απαιτούμενων υπολογισμών.

Direct Form I Transposed

## Δομές Πραγματοποίησης FIR φίλτρων

**Direct Form I- Ευθεία Μορφή I :** Προκύπτει από την άμεση εφαρμογή της γραμμικής εξίσωσης διαφορών με σταθερούς συντελεστές σε διάγραμμα βαθμίδων (transversal form)

Για φίλτρο τάξης  $K$  και με συνάρτηση μεταφοράς

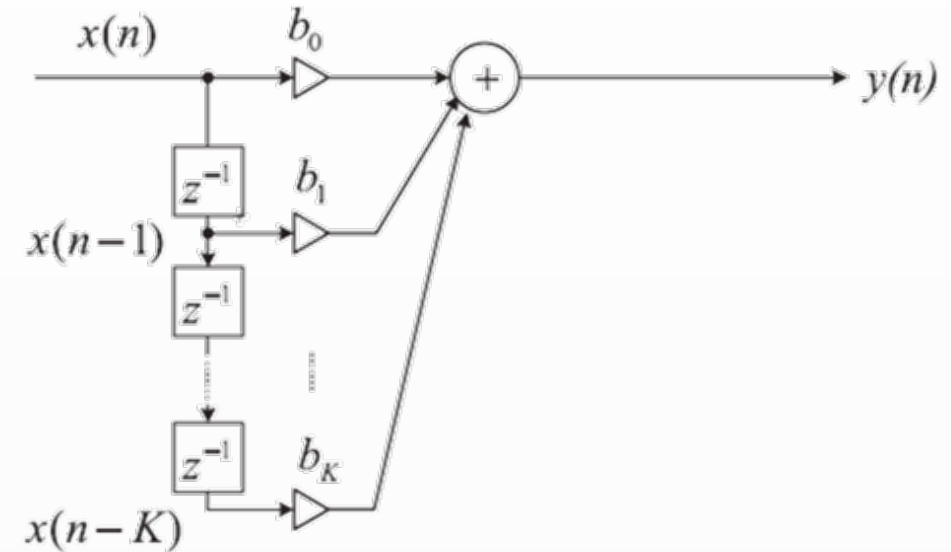
$$H(z) = \sum_{m=0}^K b_m z^{-m}$$

το πλήθος υπολογισμών στην ευθεία μορφή, είναι:

Πολλαπλασιασμοί:  $K + 1$  για κάθε δείγμα εξόδου

Προσθέσεις:  $K$  για κάθε δείγμα εξόδου

Καθυστερήσεις:  $K$



Δομή FIR συστήματος σε ευθεία μορφή

Αν υπάρχουν συμμετρίες στην κρουστική απόκριση, το πλήθος των πράξεων μπορεί να μειωθεί.

## Δομές Πραγματοποίησης FIR φίλτρων

Παράδειγμα : (α) Να σχεδιαστεί η ευθεία μορφή του FIR συστήματος με κρουστική απόκριση:

$$h[n] = \begin{cases} a^n, & 0 \leq n \leq 5 \\ 0, & \text{αλλού} \end{cases}$$

(β) Να υπολογιστεί το πλήθος των πολλαπλασιασμών και των προσθέσεων που απαιτούνται για τον υπολογισμό κάθε δείγματος εξόδου καθώς και το πλήθος των καταχωρητών καθυστέρησης.

Απάντηση: (α) Η κρουστική απόκριση γράφεται:

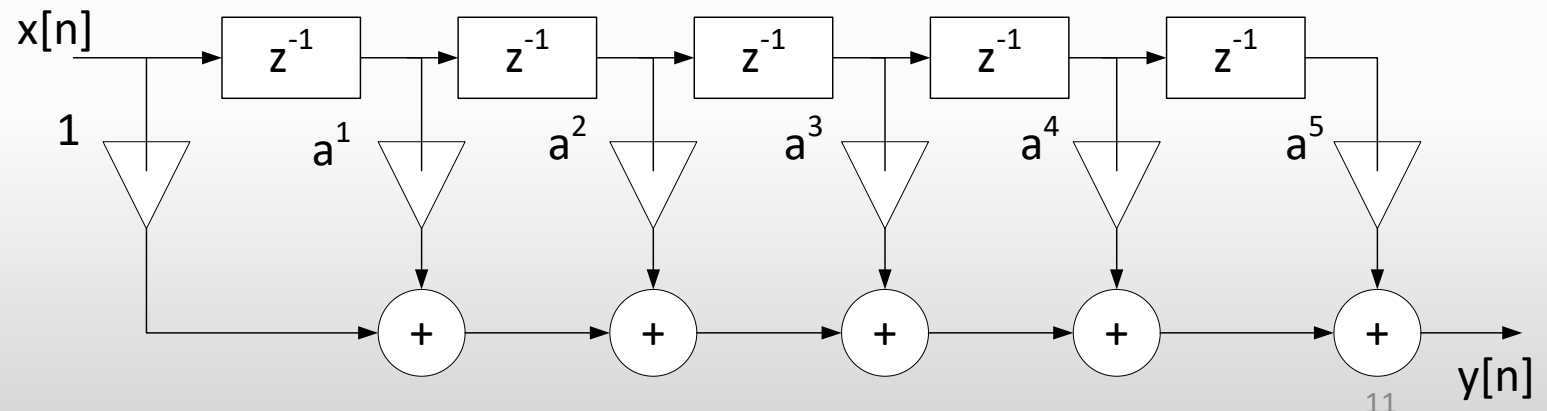
$$h[n] = a^n [u[n] - n[n - 6]] = 1 \cdot \delta(n) + a \cdot \delta(n - 1) + a^2 \cdot \delta(n - 2) + a^3 \cdot \delta(n - 3) + a^4 \cdot \delta(n - 4) + a^5 \cdot \delta(n - 5)$$

από την οποία προκύπτει ότι το διάγραμμα βαθμίδων ευθείας μορφής είναι:

(β) Από το διάγραμμα βαθμίδων προκύπτει ότι το πλήθος υπολογισμών στην ευθεία μορφή είναι:

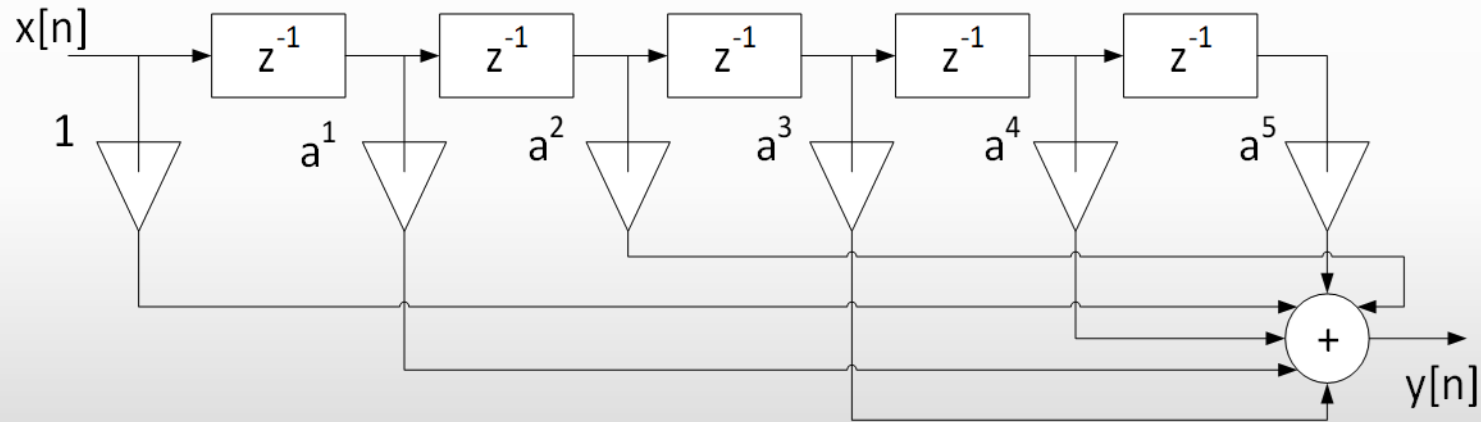
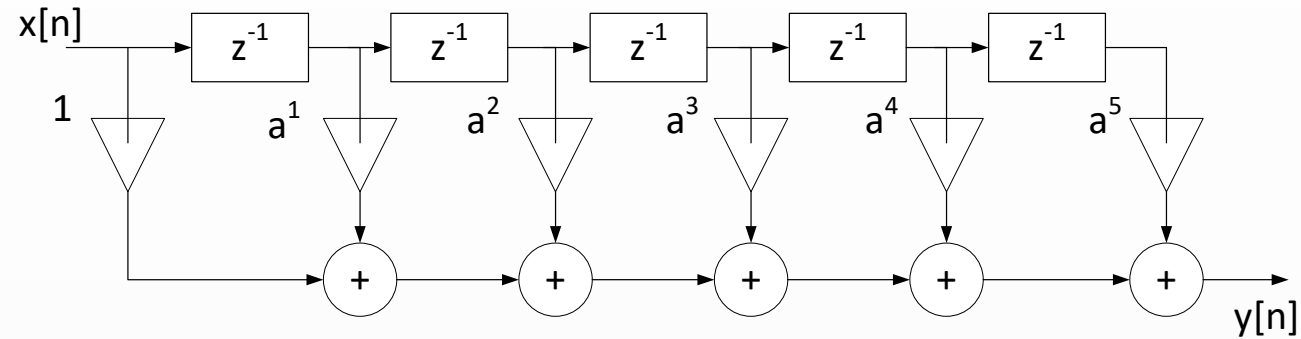
Πολλαπλασιασμοί: 6 για κάθε δείγμα εξόδου  
Προσθέσεις: 5 για κάθε δείγμα εξόδου  
Καθυστερήσεις: 5

Δομή FIR συστήματος σε ευθεία μορφή (N=5)

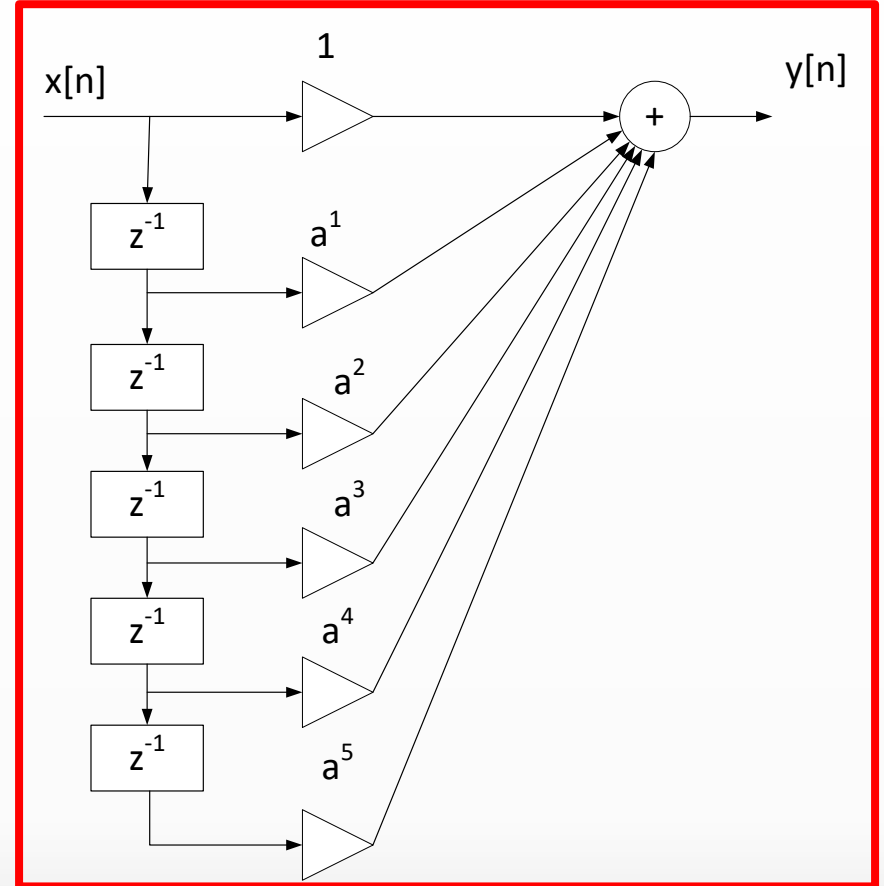


## Δομές Πραγματοποίησης FIR φίλτρων

### Tapped delay ή transversal form



### Direct Form I



## Δομές Πραγματοποίησης FIR φίλτρων

**Γραμμικής φάσης – Linear Phase Form** : Τροποποίηση της transversal μορφής λόγω της **συμμετρίας των συντελεστών** της κρουστικής απόκρισης.

Π.χ. η κρουστική απόκριση μπορεί να είναι **συμμετρική**, δηλαδή:

$$h[n] = h[K - n]$$

απλοποιούμε τη μορφή του διαγράμματος βαθμίδων.

Απαιτεί τους **~μισούς** πολλαπλασιασμούς σε σχέση με την άμεση μορφή  $K + 1$

Για φίλτρο τάξης  $K$  και με συνάρτηση μεταφοράς

$$H(z) = \sum_{m=0}^K b_m z^{-m}$$

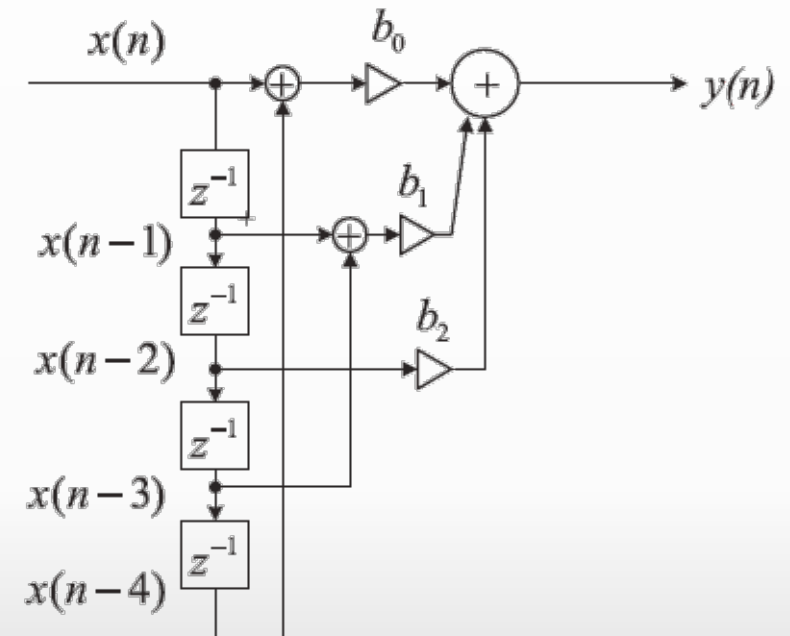
Π.χ. για  $K=4$

$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + b_1 z^{-3} + b_0 z^{-4}$$

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + b_1 x[n-3] + b_0 x[n-4]$$

$$\frac{K}{2} + 1, K \text{ άρτιος}$$

$$\frac{K+1}{2}, K \text{ περιττός}$$



Δομή FIR συστήματος σε μορφή γραμμικής φάσης

## Δομές Πραγματοποίησης FIR φίλτρων

**Μορφή Direct Form Transposed** : Τροποποίηση της transversal (Direct Form) μορφής

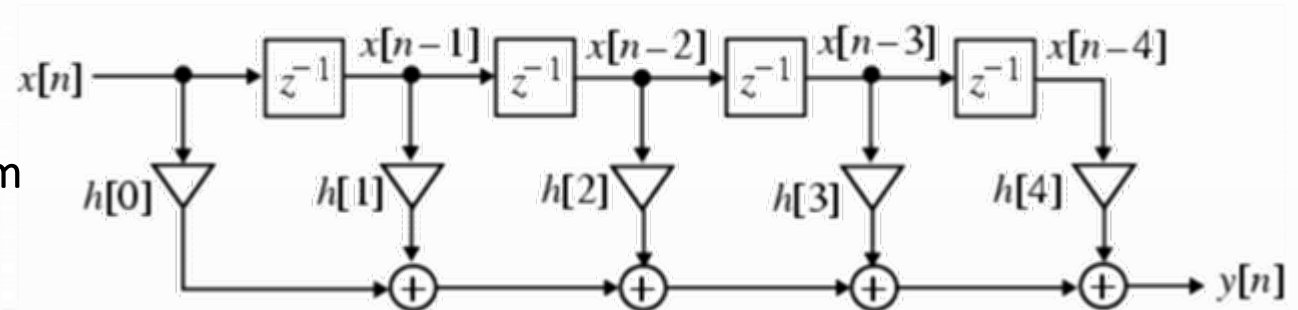
$$H(z) = \sum_{m=0}^K h_m z^{-m}, K = 4$$

Μετατροπή σε Μορφή Transpose

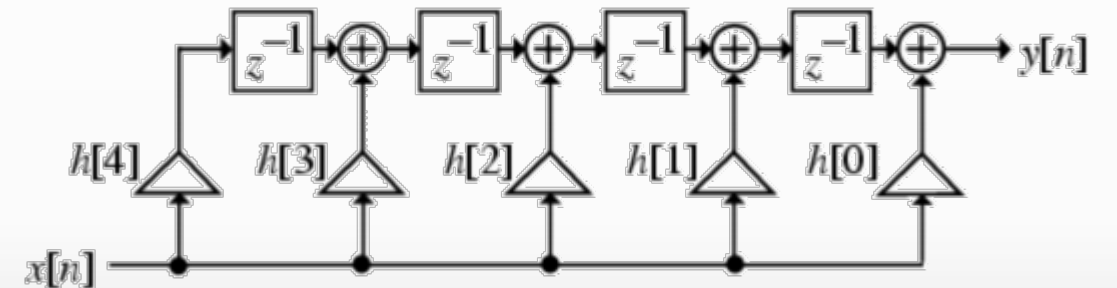
- Αντιστρέφουμε όλες τις διαδρομές
- Αντικαθιστούμε τους διακλαδωτές (pick off nodes) με αθροιστές και αντίθετα
- Αλλάζουμε την είσοδο με την έξοδο
- Η είσοδος φτάνει ταυτόχρονα σε όλους τους κλάδους άρα είναι αποδοτικότερη για παραλληλοποίηση ενώ η Direct Form είναι αποδοτικότερη για σειριακές υλοποιήσεις (loops) λόγω της μικρότερης απαίτησης σε μνήμη (λόγω data path στην DF vs του accumulator D.F. Transposed)

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + h[3]x[n-3] + h[4]x[n-4]$$

Direct Form



Transposed Form



## Δομές Πραγματοποίησης IIR φίλτρων

Η γενική μορφή της εξίσωσης διαφορών ενός IIR φίλτρου (αιτιατό αρα  $N \geq M$ ) θα είναι

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

Με συνάρτηση μεταφοράς

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}} \\ &= (b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}) \cdot \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}} \end{aligned}$$

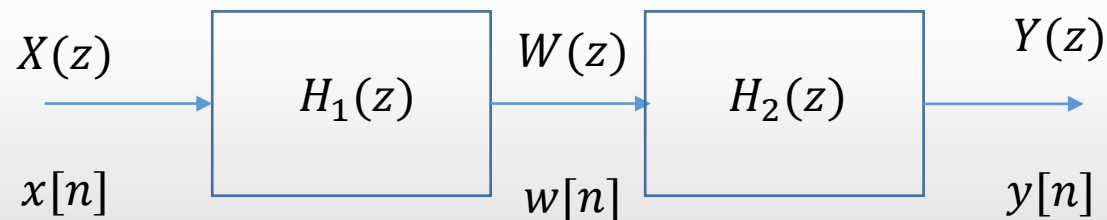
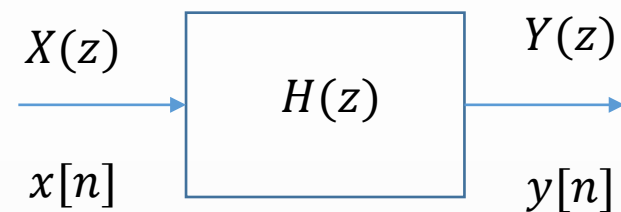
Θέτω

$$H_1(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M} = \frac{W(z)}{X(z)}$$

$$H_2(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}} = \frac{Y(z)}{W(z)}$$

$$H(z) = H_1(z) \cdot H_2(z)$$

(φίλτρο ισοδύναμο με την εν σειρά σύνδεση των φίλτρων με συναρτήσεις μεταφοράς  $H_1(z)$  και  $H_2(z)$  )



## Δομές Πραγματοποίησης IIR φίλτρων

Η γενική μορφή της εξίσωσης διαφορών ενός IIR φίλτρου (αιτιατό αρα  $N \geq M$ ) θα είναι

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

Με συνάρτηση μεταφοράς

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}} \\ &= (b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}) \cdot \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}} \end{aligned}$$

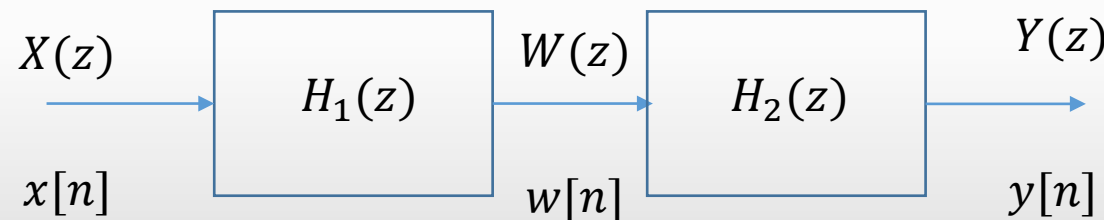
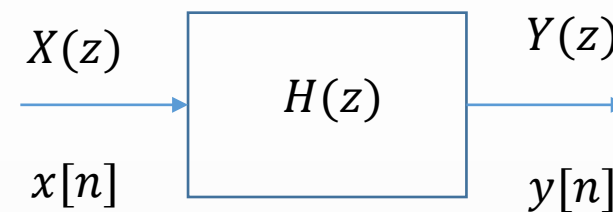
Θέτω

$$H_1(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M} = \frac{W(z)}{X(z)}$$

$$H_2(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}} = \frac{Y(z)}{W(z)}$$

$$H(z) = H_1(z) \cdot H_2(z)$$

(φίλτρο ισοδύναμο με την εν σειρά σύνδεση των φίλτρων με συναρτήσεις μεταφοράς  $H_1(z)$  και  $H_2(z)$  )



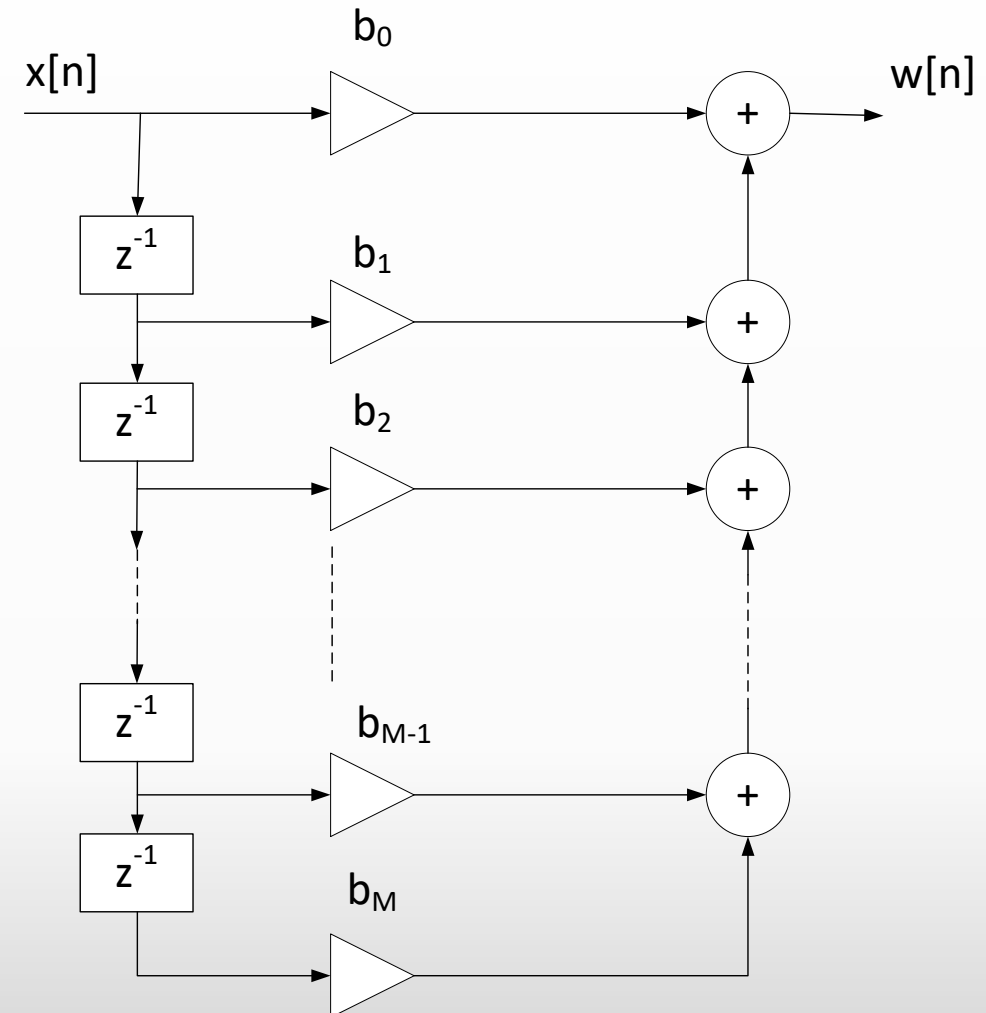


## Δομές Πραγματοποίησης IIR φίλτρων

$$H_1(z) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Mz^{-M} = \frac{W(z)}{X(z)} \rightarrow$$
$$W(z) = X(z)(b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Mz^{-M})$$

$\rightarrow$  Inverse Z Transform  $\rightarrow$

$$w[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + \dots + b_Mx[n-M]$$



## Δομές Πραγματοποίησης IIR φίλτρων

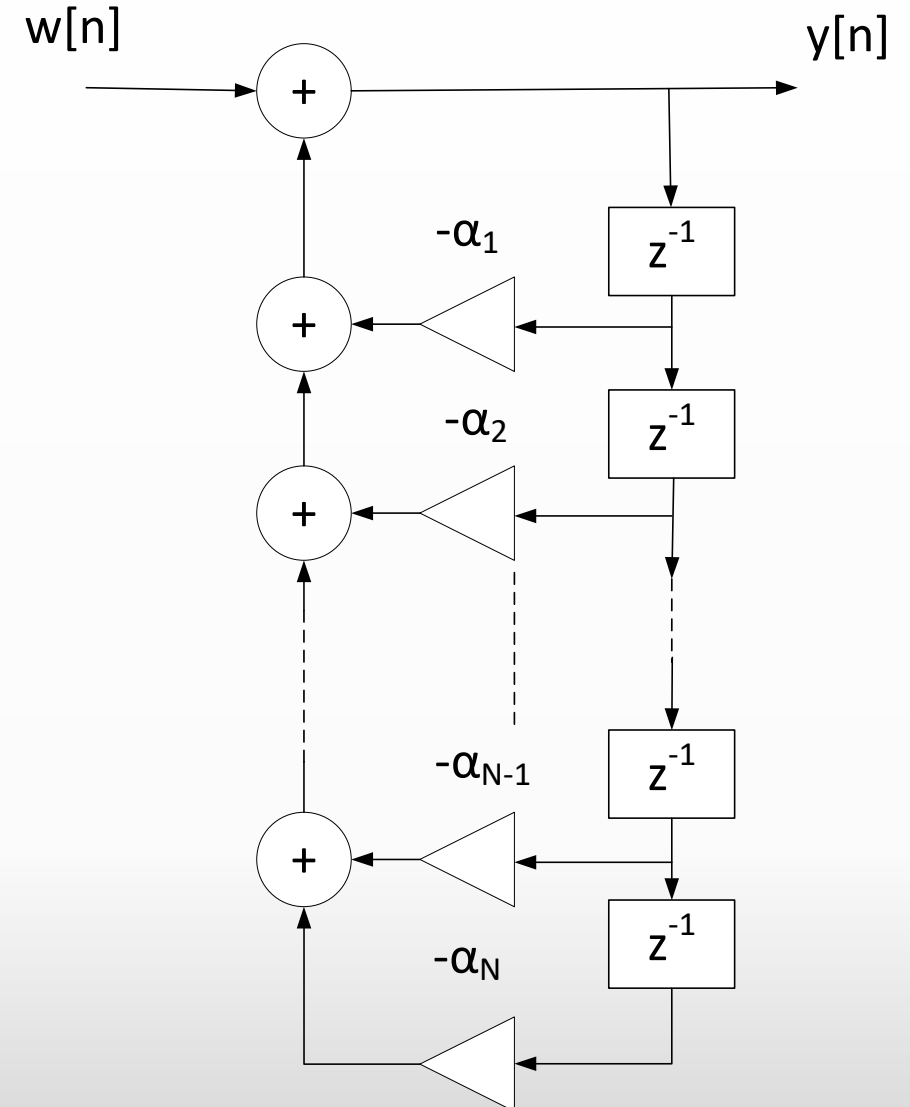
$$H_2(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}} = \frac{Y(z)}{W(z)} \rightarrow$$

$$Y(z)(1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}) = W(z)$$

$\rightarrow$  Inverse Z Transform  $\rightarrow$

$$y[n] = w[n] - a_1 y[n-1] - a_2 y[n-2] - \dots - a_N y[n-N]$$

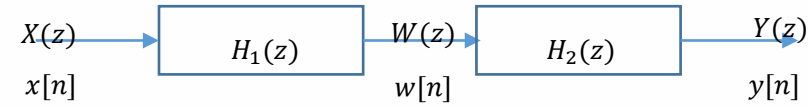
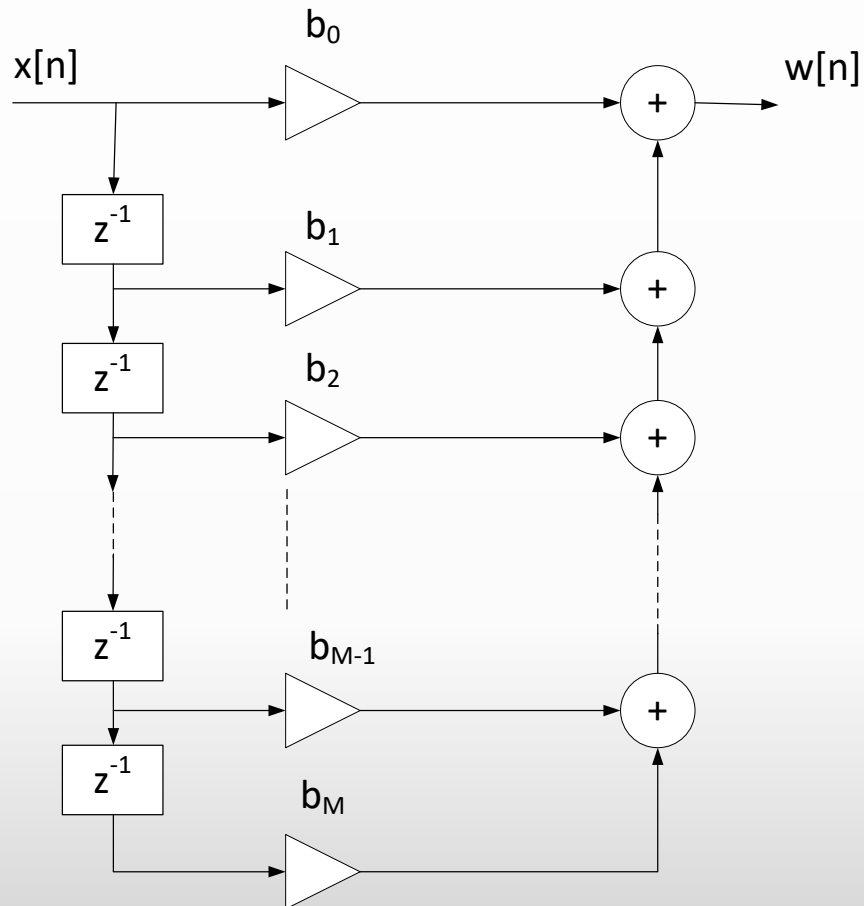
$$y[n] = w[n] - (a_1 y[n-1] + a_2 y[n-2] + \dots + a_N y[n-N])$$



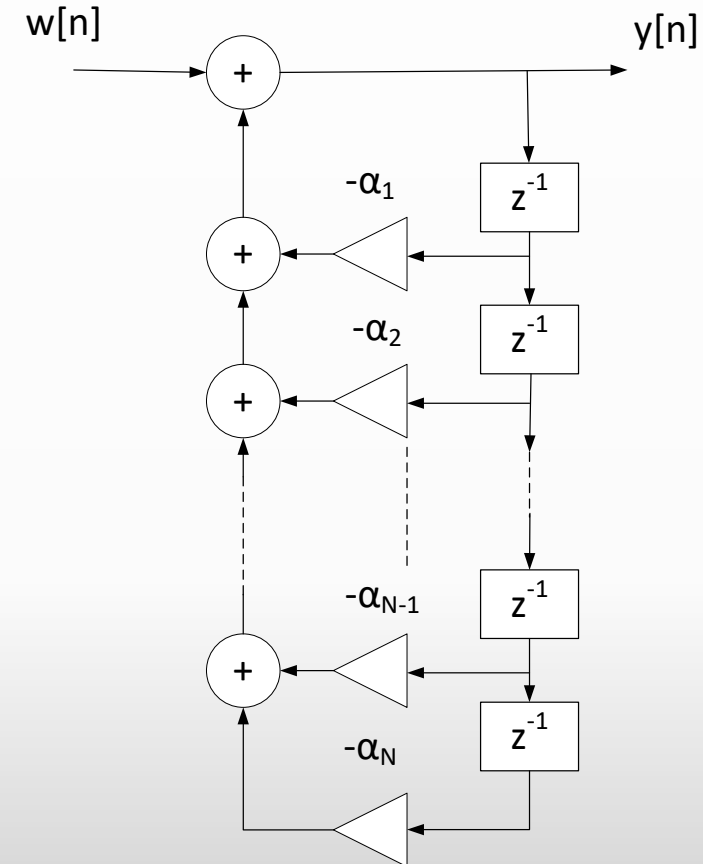
# Δομές Πραγματοποίησης IIR φίλτρων

Κάθε υποσύστημα έχει υλοποιηθεί ως μία διακλαδιζόμενη γραμμή καθυστέρησης (tapped delay line)

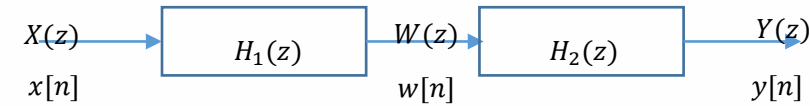
$$w[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \dots + b_M x[n-M]$$



$$y[n] = w[n] - (a_1 y[n-1] + a_2 y[n-2] + \dots + a_N y[n-N])$$



## Δομές Πραγματοποίησης IIR φίλτρων



Η γενική μορφή της εξίσωσης διαφορών ενός IIR φίλτρου (αιτιατό  $N \geq M$ ) θα είναι

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

Με συνάρτηση μεταφοράς

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}$$

Εν σειρά σύνδεση (cascade)

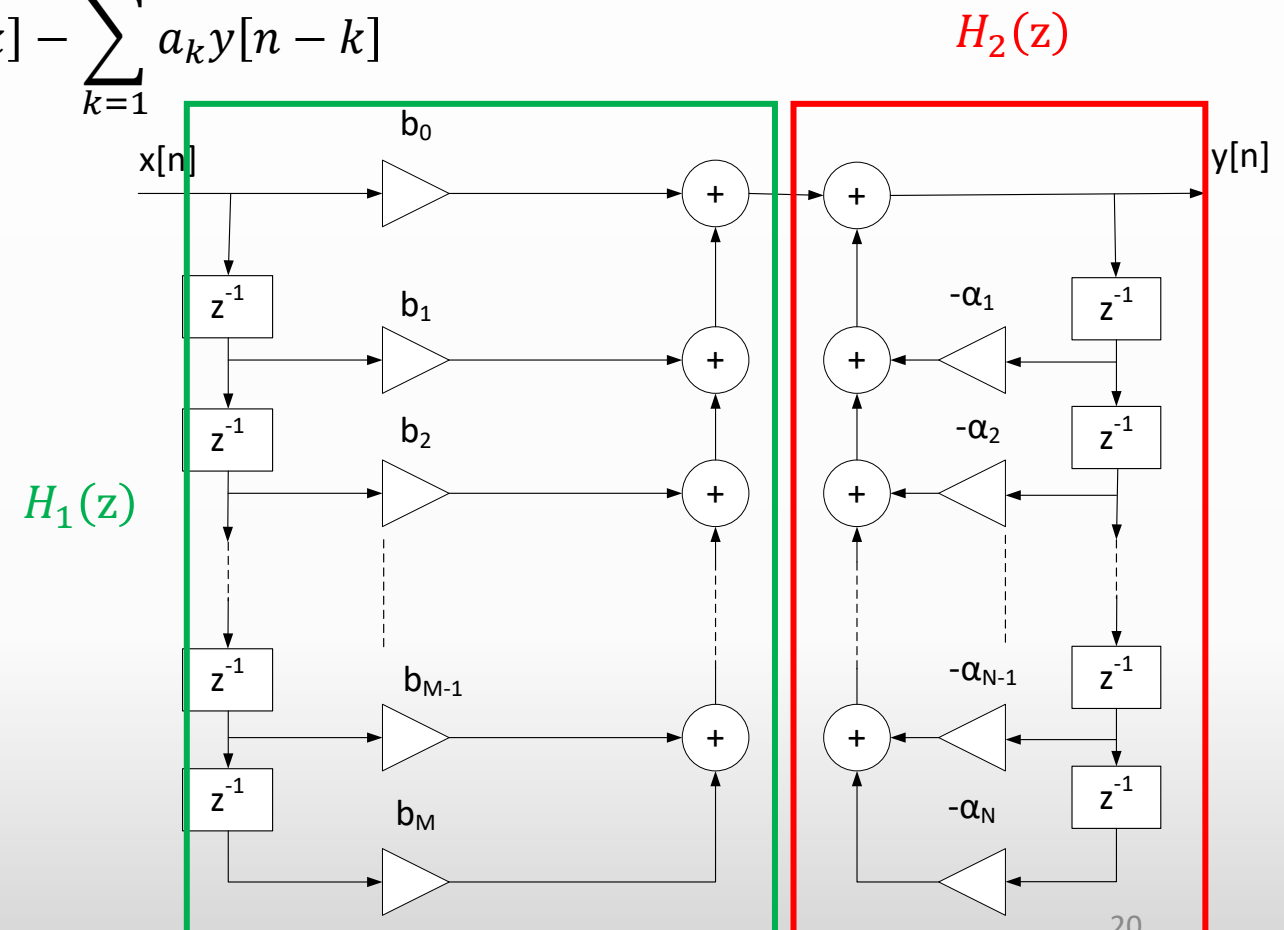
$$H(z) = H_1(z) \cdot H_2(z)$$

Direct Form I (IIR)

μη κανονική μορφή ( $N + M$  delays (max  $2N$ ) για τάξη  $N$ )

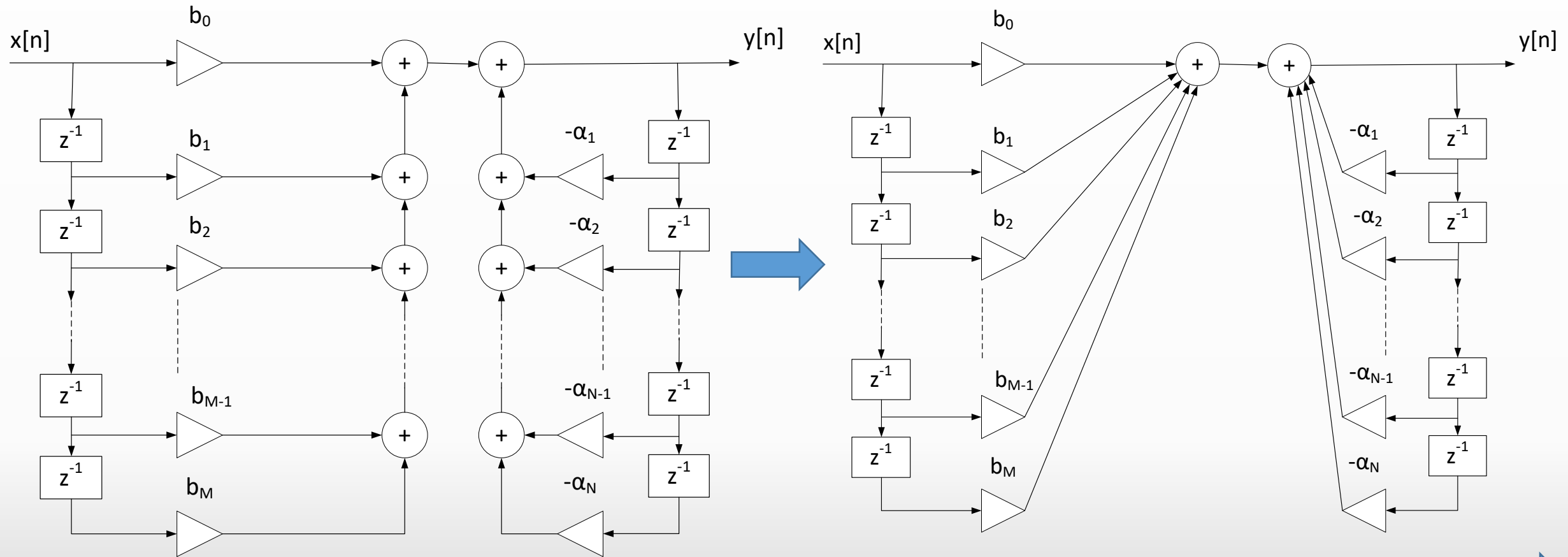
Πολλαπλασιασμοί:  $M + N + 1$

Προσθέσεις:  $M + N$



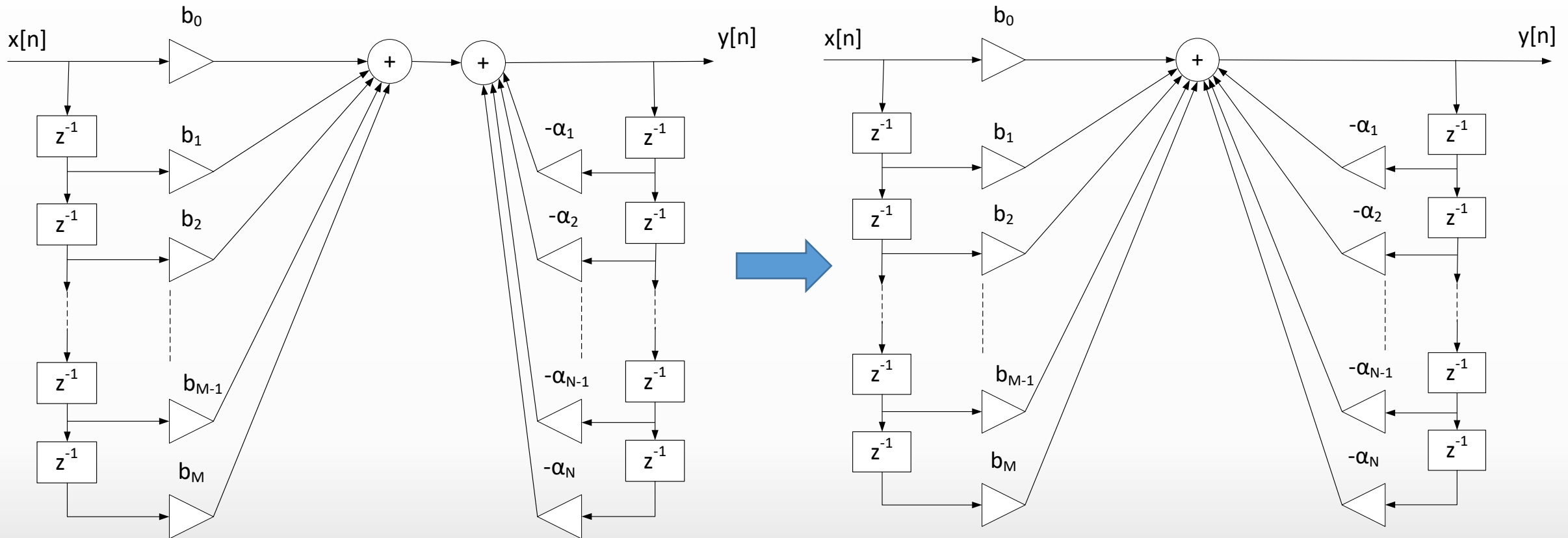
## Δομές Πραγματοποίησης IIR φίλτρων

### Direct Form I



## Δομές Πραγματοποίησης IIR φίλτρων

### Direct Form I



## Δομές Πραγματοποίησης IIR φίλτρων

Π.χ. αιτιατό ψηφιακό φίλτρο τρίτης τάξης άρα σημαίνει ότι στην εξίσωση  $N = 3$ .

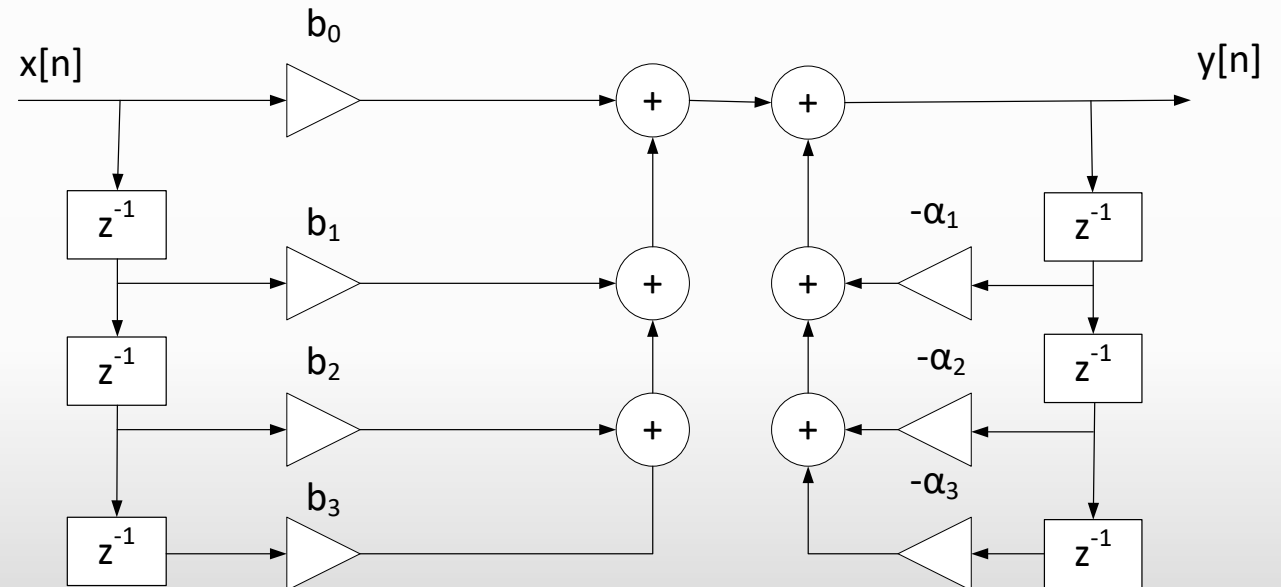
$$y[n] = b_0x[n] + b_1x[n - 1] + b_2x[n - 2] + b_3x[n - 3] - a_1y[n - 1] - a_2y[n - 2] - a_3y[n - 3]$$

η αντίστοιχη συνάρτηση μεταφοράς

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3}}{1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3}}$$

Direct Form I (IIR)

μη κανονική μορφή (6 delays για τάξη 3)



## Δομές Πραγματοποίησης IIR φίλτρων

Η γενική μορφή της εξίσωσης διαφορών ενός IIR φίλτρου (αιτιατό  $N \geq M$ ) θα είναι

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

Με συνάρτηση μεταφοράς

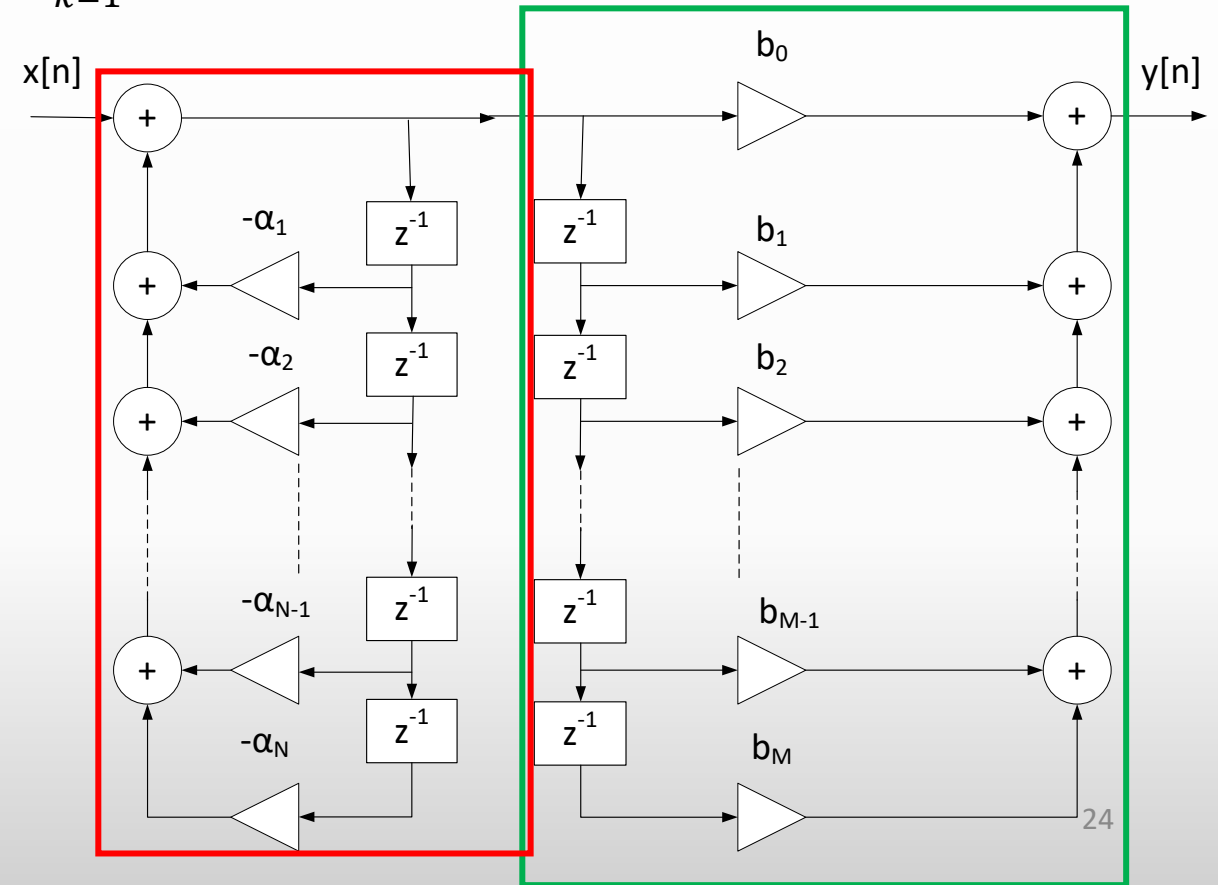
$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}$$

Εν σειρά σύνδεση (cascade)

$$H(z) = H_1(z) \cdot H_2(z)$$

Αντιμετάθεση  $\rightarrow$  αντιμετάθεση των υποσυστημάτων

$$H(z) = H_1(z) \cdot H_2(z) = H_2(z) \cdot H_1(z)$$





## Δομές Πραγματοποίησης IIR φίλτρων

Η γενική μορφή της εξίσωσης διαφορών ενός IIR φίλτρου (αιτιατό  $N \geq M$ ) θα είναι

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

## Με συνάρτηση μεταφοράς

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_o + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}$$

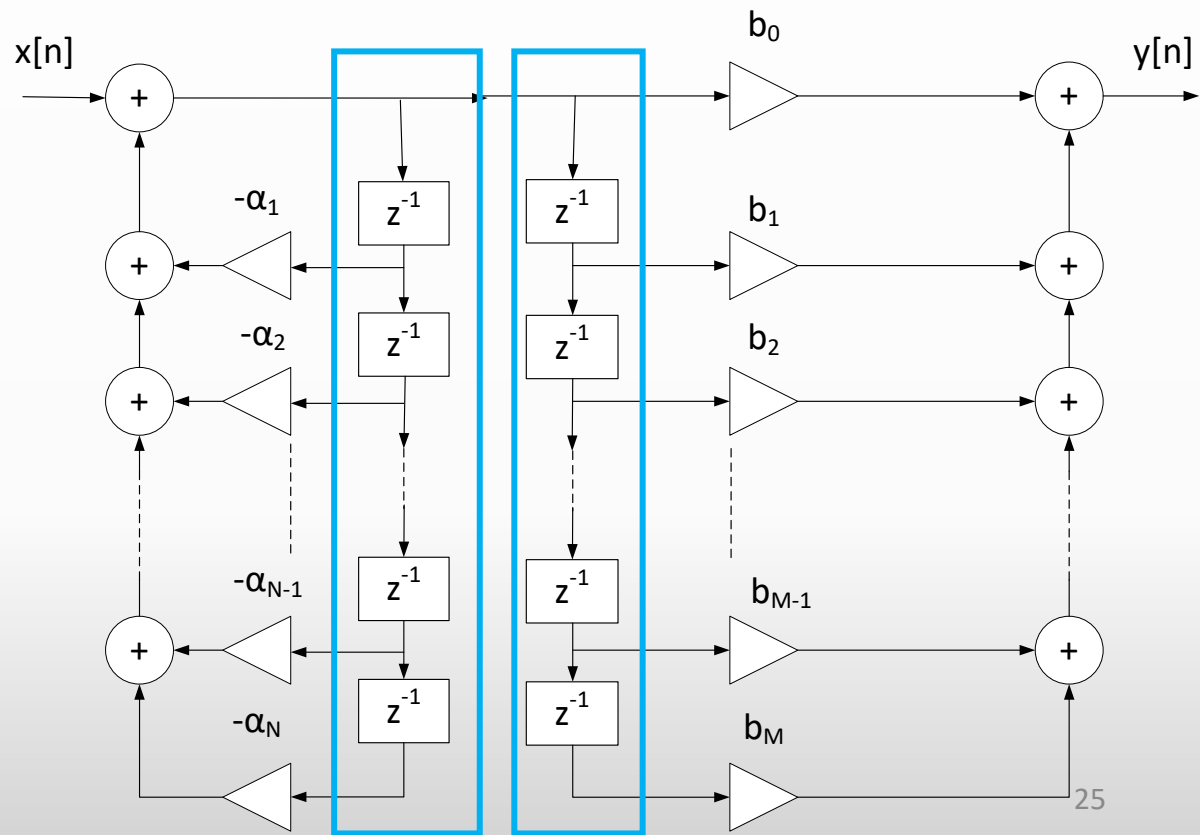
## Εν σειρά σύνδεση (cascade)

$$H(z) = H_1(z) \cdot H_2(z)$$

## Αντιμετάθεση

$$H(z) = H_1(z) \cdot H_2(z) = H_2(z) \cdot H_1(z)$$

- Εάν  $N=M$ , οι γραμμές καθυστέρησης είναι ίδιες άρα μπορεί χρησιμοποιηθεί **μία**
- προφανώς ισχύει και γενικά



## Δομές Πραγματοποίησης IIR φίλτρων

Η γενική μορφή της εξίσωσης διαφορών ενός IIR φίλτρου (αιτιατό  $N \geq M$ ) θα είναι

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

Με συνάρτηση μεταφοράς

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}$$

Εν σειρά σύνδεση (cascade)

$$H(z) = H_1(z) \cdot H_2(z)$$

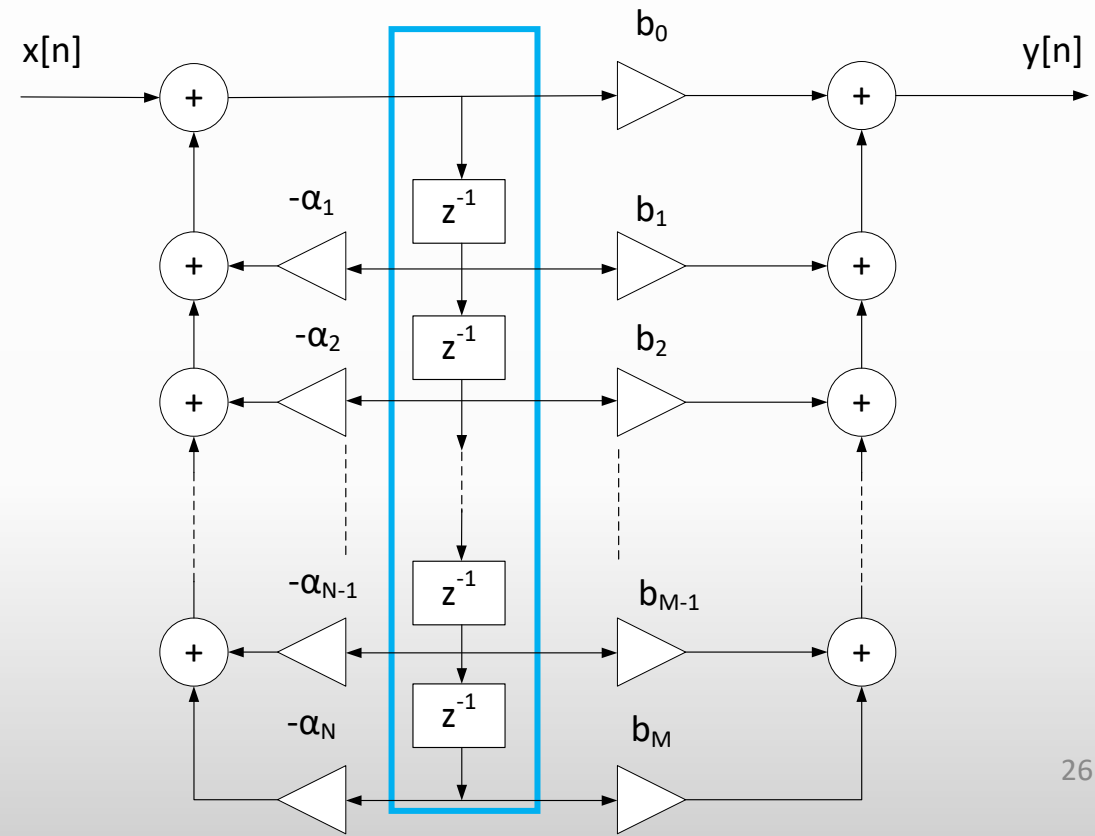
Αντιμετάθεση

$$H(z) = H_1(z) \cdot H_2(z) = H_2(z) \cdot H_1(z)$$

Direct Form II (IIR) ή

«Απευθείας μορφή πρώτα ο παρονομαστής»

Κανονική (canonical) μορφή (N delays για τάξη N)



## Δομές Πραγματοποίησης IIR φίλτρων

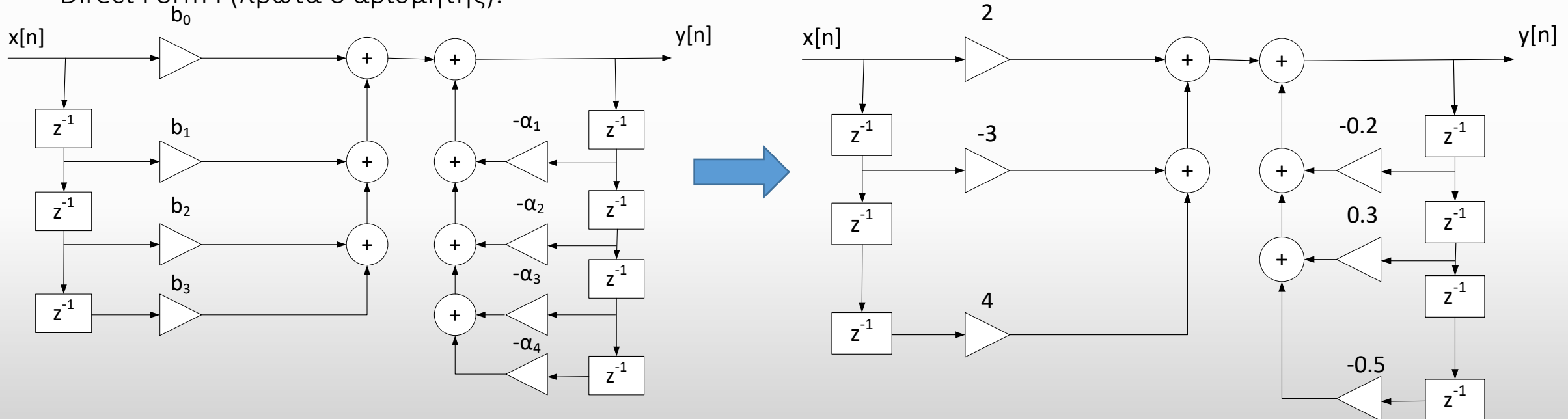
Παράδειγμα : Για το φίλτρο με συνάρτηση μεταφοράς  $H(z) = \frac{2-3z^{-1}+4z^{-3}}{1+0.2z^{-1}-0.3z^{-2}+0.5z^{-4}}$

να υπολογιστεί η εξίσωση διαφορών και να σχεδιαστούν η Direct Form I και η Direct Form II

- Εξίσωση διαφορών

$$y[n] = 2x[n] - 3x[n-1] + 4x[n-3] - 0.2y[n-1] + 0.3y[n-2] - 0.5y[n-4]$$

- Direct Form I (πρώτα ο αριθμητής):



## Δομές Πραγματοποίησης IIR φίλτρων

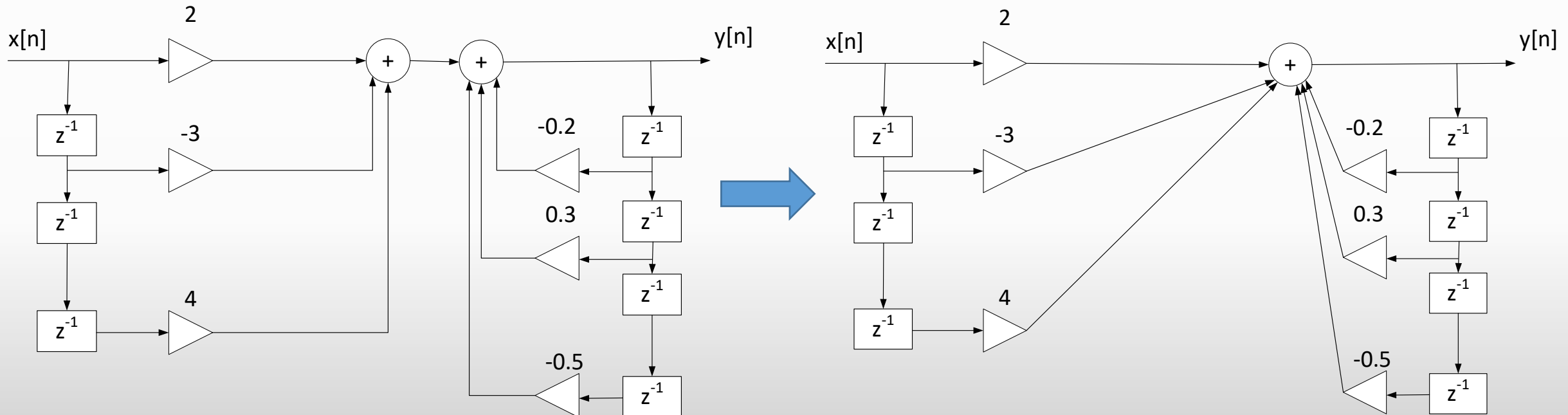
Παράδειγμα(συνέχεια) : Για το φίλτρο με συνάρτηση μεταφοράς  $H(z) = \frac{2-3z^{-1}+4z^{-3}}{1+0.2z^{-1}-0.3z^{-2}+0.5z^{-4}}$

να σχεδιαστεί η Direct Form I, η Direct Form II και να υπολογιστούν οι αντίστοιχες εξισώσεις διαφορών

- Εξίσωση διαφορών

$$y[n] = 2x[n] - 3x[n-1] + 4x[n-3] - 0.2y[n-1] + 0.3y[n-2] - 0.5y[n-4]$$

- Direct Form I (πρώτα ο αριθμητής):

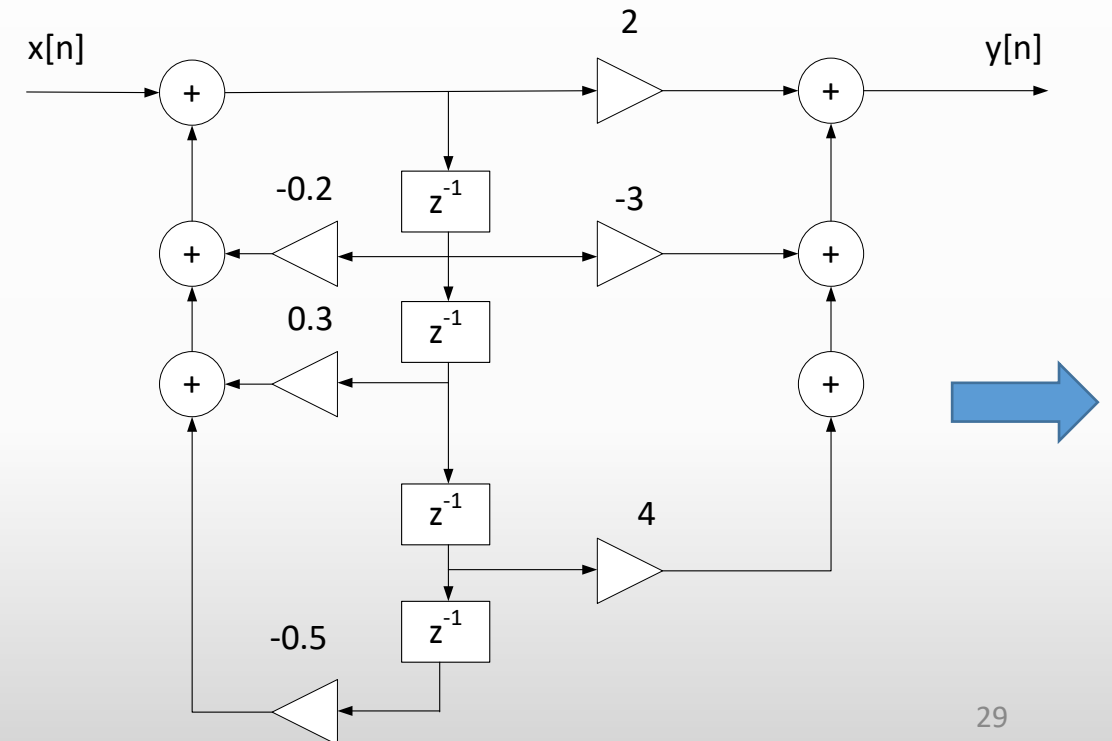
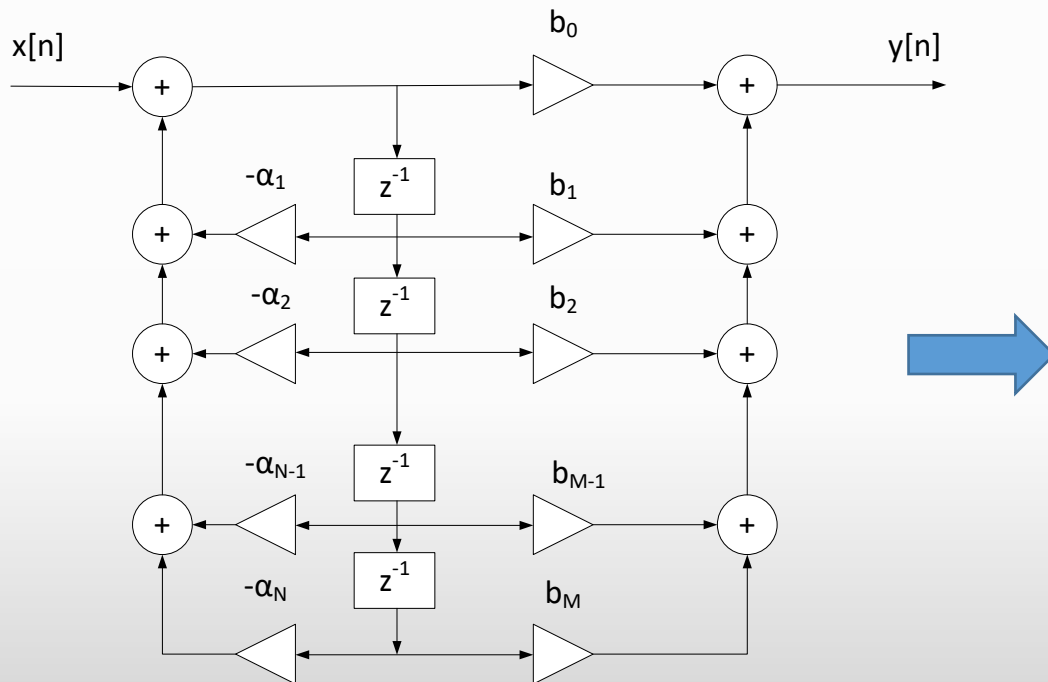


## Δομές Πραγματοποίησης IIR φίλτρων

Παράδειγμα(συνέχεια) : Για το φίλτρο με συνάρτηση μεταφοράς  $H(z) = \frac{2-3z^{-1}+4z^{-3}}{1+0.2z^{-1}-0.3z^{-2}+0.5z^{-4}}$

να σχεδιαστεί η Direct Form I, η Direct Form II και να υπολογιστούν οι αντίστοιχες εξισώσεις διαφορών

Direct Form II (πρώτα ο παρονομαστής):  $\max(M,N)=(3,4)=4$

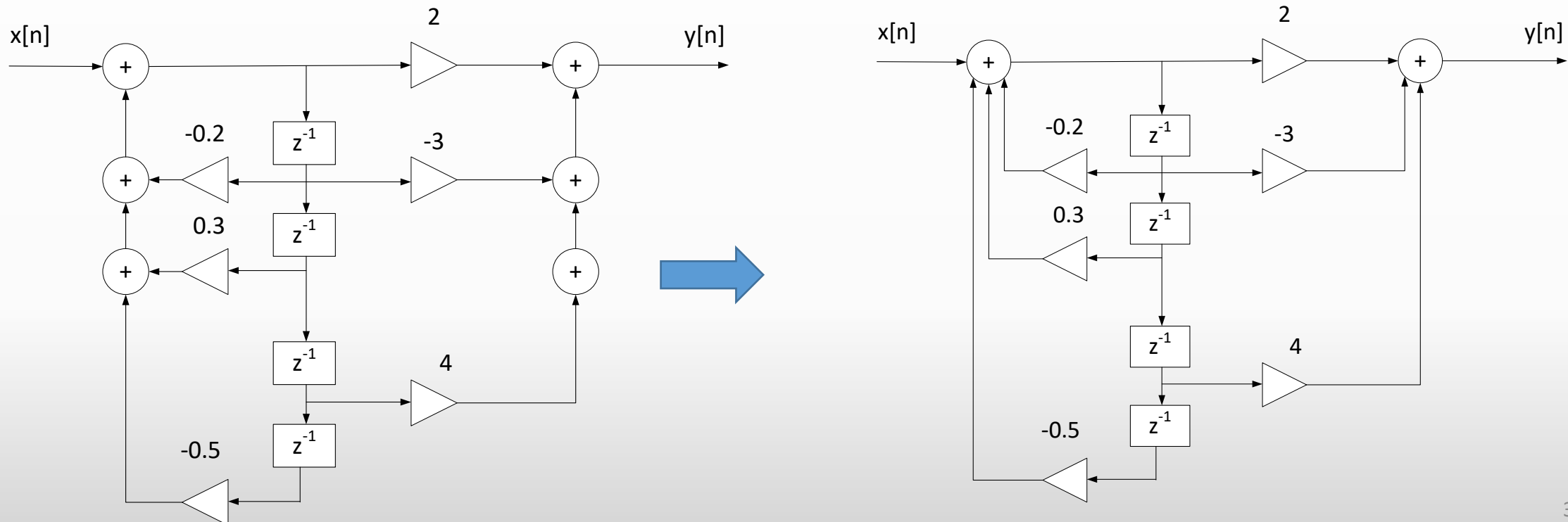


## Δομές Πραγματοποίησης IIR φίλτρων

Παράδειγμα(συνέχεια) : Για το φίλτρο με συνάρτηση μεταφοράς  $H(z) = \frac{2-3z^{-1}+4z^{-3}}{1+0.2z^{-1}-0.3z^{-2}+0.5z^{-4}}$

να σχεδιαστεί η Direct Form I, η Direct Form II και να υπολογιστούν οι αντίστοιχες εξισώσεις διαφορών

Direct Form II (πρώτα ο παρονομαστής):  $\max(M,N)=(3,4)=4$



## Δομές Πραγματοποίησης IIR φίλτρων

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

Η ΓΕΔΣΣ των IIR μπορεί να διασπαστεί σε δύο σχέσεις ως εξής:

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

Direct Form I - πρώτα πόλοι

Η πρώτη σχέση περιγράφει ένα υποσύστημα με είσοδο το σήμα  $x[n]$  και έξοδο το  $w[n]$ . Από τη μορφή της σχέσης προκύπτει ότι το υποσύστημα αυτό είναι **μόνο πόλων** (all-pole).

Η δεύτερη σχέση περιγράφει ένα FIR υποσύστημα που δέχεται ως είσοδο το σήμα  $w[n]$  και παράγει την έξοδο  $y[n]$  του συνολικού συστήματος. Το υποσύστημα αυτό υλοποιεί τα **μηδενικά** του συστήματος.

Τα δύο υποσυστήματα είναι συνδεδεμένα σε σειρά και στο πεδίο της συχνότητας περιγράφονται από τις ακόλουθες σχέσεις:

$$W(z) = [b_0 + b_1 z^{-1} + \dots + b_M z^{-M}] X(z)$$
$$Y(z) = W(z) - [a_1 z^{-1} + \dots + a_N z^{-N}] Y(z)$$

## Δομές Πραγματοποίησης IIR φίλτρων

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

Με βάση την αντιμεταθετική ιδιότητα εναλλάσσουμε αμοιβαία τη σειρά των υποσυστημάτων της ευθείας μορφής I, οπότε η ΓΕΔΣΣ γράφεται:

$$w[n] = x[n] - \sum_{k=1}^N a_k w[n-k]$$

$$y[n] = \sum_{m=0}^M b_m w[n-m]$$

Direct Form II - πρώτα μηδενικά

Οι παραπάνω σχέσεις στο πεδίο της συχνότητας γράφονται:

$$W(z) = X(z) - [a_1 z^{-1} + \dots + a_N z^{-N}] W(z)$$

$$Y(z) = [b_0 + b_1 z^{-1} + \dots + b_M z^{-M}] W(z)$$

Με τη διαδικασία αυτή υλοποιούνται πρώτα τα μηδενικά και κατόπιν οι πόλοι του συστήματος.



## Δομές Πραγματοποίησης IIR φίλτρων

Η γενική μορφή της εξίσωσης διαφορών ενός IIR φίλτρου (αιτιατό  $N \geq M$ ) θα είναι

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

Με συνάρτηση μεταφοράς

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}$$

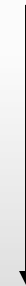
Στη πράξη για να μειωθούν τα σφάλματα των υπολογισμών κατά την υλοποίησή τους με ψηφιακά συστήματα τα οποία χρησιμοποιούν πεπερασμένη ακρίβεια για την αναπαράσταση των συντελεστών και τον υπολογισμό των πράξεων

οι υψηλής τάξης συναρτήσεις ( $N > 2$ ) πραγματοποιούνται ως συνδυασμοί δομών πρώτης ή/και δεύτερης τάξης σε σειρά (cascade) ή παράλληλα (parallel) :

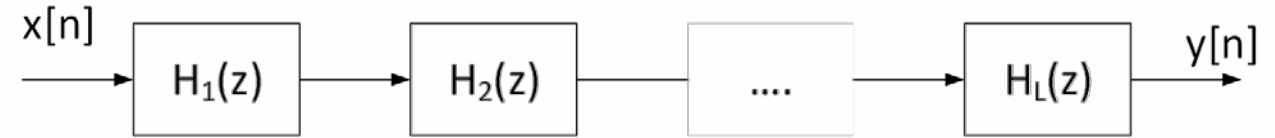
$$H(z) = H_1(z) \cdot H_2(z) \cdot \dots \cdot H_L(z) \quad (\text{cascade})$$

ή

$$H(z) = H_1(z) + H_2(z) + \dots + H_L(z) \quad (\text{parallel})$$



## Δομές Πραγματοποίησης IIR φίλτρων



Για την πραγματοποίηση σε σειρά ή μορφή καταρράκτη (cascade realization), η συνάρτηση μεταφοράς  $H(z)$  αναλύεται σε γινόμενο παραγόντων ως:

$$H(z) = H_1(z) \cdot H_2(z) \cdot \dots \cdot H_L(z) = \prod_{i=1}^L H_i(z)$$

$L$  ακέραιο μέρος του  $\frac{N+1}{2}$  και  $H_i(z)$  είναι δομή πρώτης ή δεύτερης τάξης:

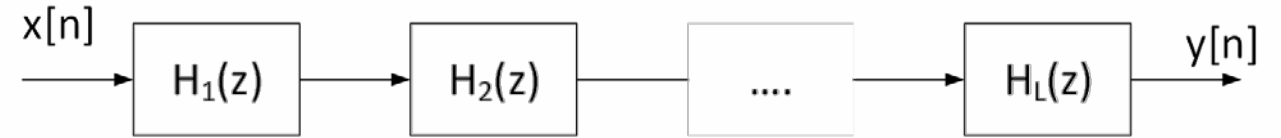
$$H_i(z) = \frac{b_{0i} + b_{1i}z^{-1}}{1 + a_{1i}z^{-1}} \quad (1\text{ης τάξης})$$

$$H_i(z) = \frac{b_{0i} + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \quad (2\text{ης τάξης})$$

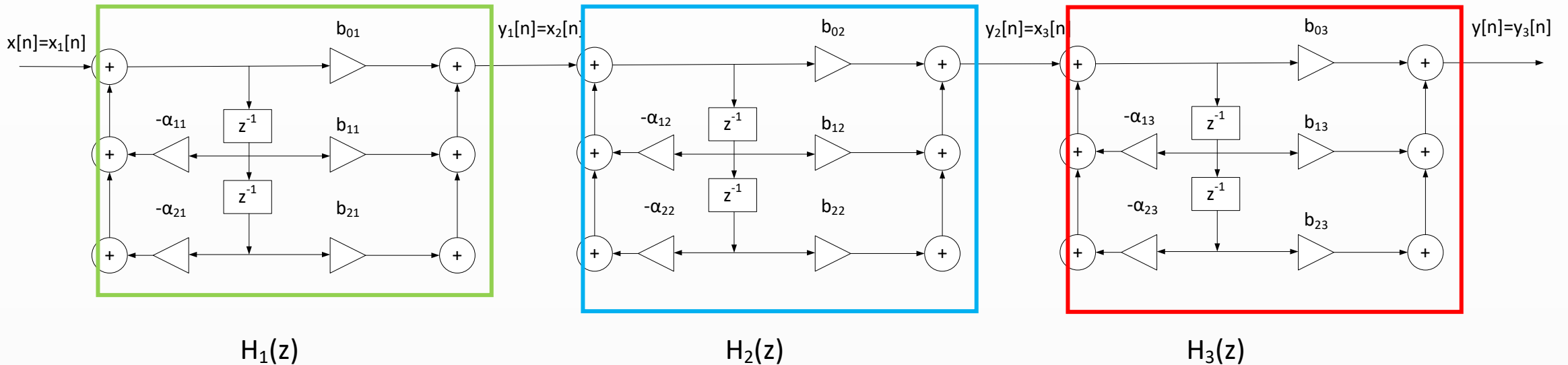
Τρία είναι τα προβλήματα που πρέπει να αντιμετωπιστούν κατά την πραγματοποίηση σε σειρά:

- με ποιο τρόπο θα πρέπει να συνδυαστούν οι αριθμητές με τους παρονομαστές, μετά την παραγοντοποίηση της  $H(z)$
- με ποια σειρά θα πρέπει οι επιμέρους δομές να συνδεθούν, και
- η ανάγκη κλιμάκωσης, δηλαδή μείωσης του πλάτους του σήματος σε ενδιάμεσα σημεία της δομής, ώστε αυτό να μην είναι ούτε πολύ μεγάλο, ούτε πολύ μικρό.

## Δομές Πραγματοποίησης IIR φίλτρων



πραγματοποίηση σε σειρά (cascade realization),  $H(z) = H_1(z) \cdot H_2(z) \cdot \dots \cdot H_L(z) = \prod_{i=1}^L H_i(z)$ , έως 2<sup>ης</sup> τάξης και  $H_i(z)$  παρουσιάζεται στην Direct Form II (canonical)



$$H(z) = H_1(z) \cdot H_2(z) \cdot H_3(z)$$

Second Order Sections (SOS)

Για τον υπολογισμό στο Matlab των συντελεστών της μορφής καταρράκτη : συνάρτηση `tf2sos()`.

## Δομές Πραγματοποίησης IIR φίλτρων

Για την παράλληλη πραγματοποίηση (**parallel realization**), η συνάρτηση μεταφοράς  $H(z)$  αναλύεται σε άθροισμα παραγόντων ως:

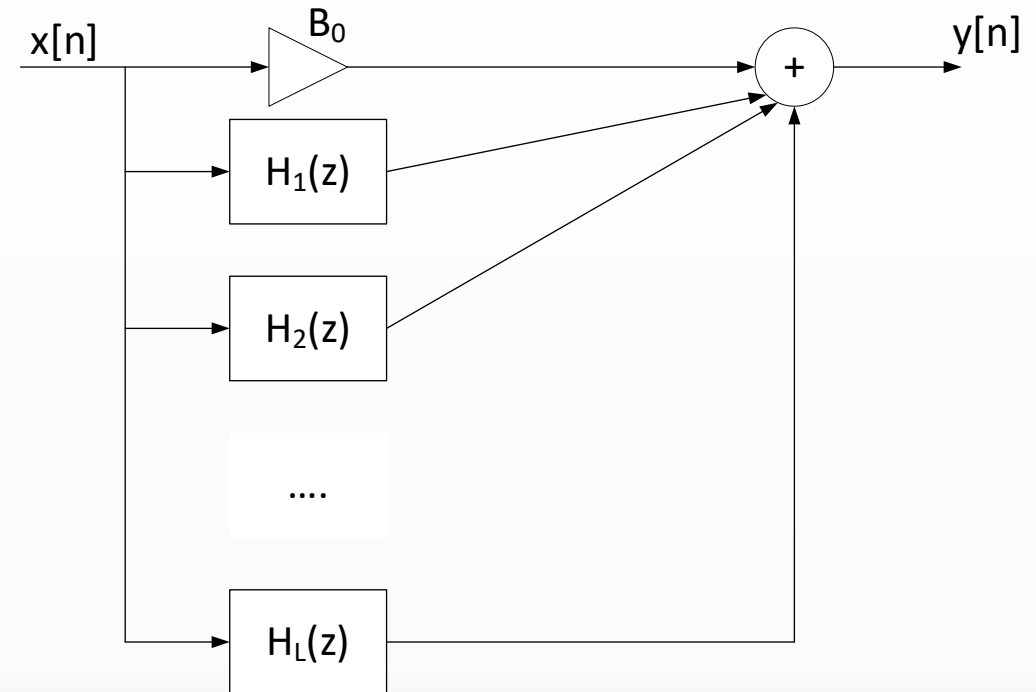
$$H(z) = B_0 + H_1(z) + H_2(z) + \dots + H_L(z) = B_0 + \sum_{i=1}^L H_i(z)$$

$L$  ακέραιο μέρος του  $\frac{N+1}{2}$ ,  $B_0 = \frac{b_M}{a_N}$

και  $H_i(z)$  είναι δομή πρώτης ή δεύτερης τάξης και

$$H_i(z) = \frac{b_{oi} + b_{1i}z^{-1}}{1 + a_{1i}z^{-1}} \quad (1\text{ης τάξης})$$

$$H_i(z) = \frac{b_{oi} + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \quad (2\text{ης τάξης})$$



- Στην παράλληλη πραγματοποίηση δεν παίζει ρόλο η σειρά διασύνδεσης των επιμέρους δομών.
- Επιπλέον, η κλιμάκωση είναι πιο εύκολη γιατί μπορεί να γίνει για κάθε μία δομή ανεξάρτητα.
- Τα μηδενικά των επιμέρους δομών είναι περισσότερο ευαίσθητα στα σφάλματα κβάντισης των συντελεστών.

## Επίδραση Πεπερασμένου Μήκους Λέξης στην Ακρίβεια των Φίλτρων

παρατηρήσεις:

- Ένα IIR φίλτρο είναι ευάλωτο στην ακρίβεια απεικόνισης των συντελεστών του, επειδή μικρές μεταβολές στην αναπαράσταση των συντελεστών μπορεί να δημιουργήσει μεγάλη αλλαγή στην απόκριση συχνότητας, καθώς είναι δυνατό οι πόλοι του φίλτρου να μετακινηθούν επάνω ή εκτός του μοναδιαίου κύκλου, οπότε το φίλτρο καθίσταται ασταθές.
- Ένα FIR φίλτρο δεν διαθέτει πόλους, οπότε η μειωμένη ακρίβεια απεικόνισης των συντελεστών του μπορεί να το καθιστά λιγότερο βέλτιστο, δεν το καθιστά όμως ασταθές.
- Το γεγονός ότι τα FIR φίλτρα είναι πάντα ευσταθή είναι ένα σημαντικό χαρακτηριστικό τους.
- Ένα άλλο πλεονέκτημά τους είναι ότι έχουν πάντα γραμμική απόκριση φάσης.
- Αποδεικνύεται ότι οι επιπτώσεις της πεπερασμένης ακρίβειας στην αναπαράσταση των συντελεστών και στον υπολογισμό των πράξεων σε ένα ψηφιακό φίλτρο, μπορούν να μειωθούν αν το φίλτρο με τάξη υψηλότερης της δεύτερης μπορεί να υλοποιηθεί ως συνδυασμός δομών πρώτης και/ή δεύτερης τάξης σε σειρά ή παράλληλα.

## Πραγματοποίηση & Υλοποίηση Ψηφιακών Φίλτρων

- Είδαμε τεχνικές σχεδίασης ψηφιακών φίλτρων και μάθαμε να προσδιορίζουμε τη συνάρτηση μεταφοράς και την εξίσωση διαφορών LTI συστημάτων διακριτού χρόνου. Το τελευταίο στάδιο είναι η υλοποίηση των φίλτρων αυτών, ώστε να επιτελέσουν το σκοπό για τον οποίο αυτά σχεδιάστηκαν.
- Η υλοποίηση των IIR φίλτρων, όπως και αυτή των FIR φίλτρων, μπορεί να γίνει είτε σε κυκλωματική μορφή (hardware), είτε με προγραμματισμό (software) ενός υπολογιστικού συστήματος γενικού ή ειδικού σκοπού.
- Γενικά, η εξίσωση διαφορών ενός φίλτρου είναι μια υπολογιστική διαδικασία (ένας αλγόριθμος) για τον υπολογισμό της ακολουθίας εξόδου  $y(n)$  του συστήματος, από την ακολουθία εισόδου  $x(n)$ .
- οι υπολογισμοί αυτοί μπορούν να γίνουν με ένα διαφορετικό, αλλά ισοδύναμο, σύνολο εξισώσεων διαφορών.
- Κάθε ένα τέτοιο σύνολο εξισώσεων ορίζει μια υπολογιστική διαδικασία ή έναν αλγόριθμο για την υλοποίηση του συστήματος. Για κάθε ένα σύνολο εξισώσεων μπορούμε να σχεδιάσουμε ένα διάγραμμα βαθμίδων το οποίο να αποτελείται από στοιχεία καθυστέρησης, πολλαπλασιαστές και αθροιστές. → δομή πραγματοποίησης (realization structure) του συστήματος.
- Στην περίπτωση κατά την οποία το σύστημα (φίλτρο) υλοποιηθεί προγραμματιστικά (software), το διάγραμμα βαθμίδων ή ισοδύναμο το σύνολο των εξισώσεων διαφορών, μετατρέπεται σε ένα σύνολο εντολών (πρόγραμμα) το οποίο «τρέχει» στο συγκεκριμένο υπολογιστικό σύστημα.

## Πραγματοποίηση & Υλοποίηση Ψηφιακών Φίλτρων

- Υπάρχουν διαφορετικές δομές πραγματοποίησης ενός φίλτρου, για παράδειγμα, άμεση μορφή τύπου I ή II, σε σειρά ή παράλληλη γιατί οι διάφορες δομές πραγματοποίησης παρουσιάζουν διαφορετικά χαρακτηριστικά ως προς την απαιτούμενη υπολογιστική πολυπλοκότητα (computational complexity), τη μνήμη (memory) και τα προβλήματα λόγω του πεπερασμένου μήκους λέξης (finite wordlength effects).
- υπολογιστική πολυπλοκότητα** : το πλήθος των αριθμητικών πράξεων (πολλαπλασιασμών, διαιρέσεων και προσθέσεων) οι οποίες απαιτούνται για τον υπολογισμό κάθε δείγματος εξόδου  $y(n)$  καθώς και το πλήθος των προσπελάσεων της μνήμης ή το πλήθος των συγκρίσεων δύο αριθμών.
- μνήμη** : το πλήθος των θέσεων μνήμης που απαιτούνται για την αποθήκευση των δειγμάτων εισόδου και εξόδου, των παραμέτρων του συστήματος, καθώς και των ενδιάμεσων αποτελεσμάτων που τυχόν προκύπτουν.
- πεπερασμένο μήκος λέξης (πεπερασμένη ακρίβεια)** : φαινόμενα κβάντισης τα οποία ενυπάρχουν σε οποιαδήποτε ψηφιακή υλοποίηση ενός συστήματος είτε κυκλωματική, είτε προγραμματιστική.

## Πραγματοποίηση & Υλοποίηση Ψηφιακών Φίλτρων

- Οι παράμετροι του συστήματος πρέπει απαραίτητα να αναπαρασταθούν με πεπερασμένη ακρίβεια. Τα αποτελέσματα που προκύπτουν στη διαδικασία υπολογισμού κάθε δείγματος εξόδου του συστήματος, πρέπει να στρογγυλευθούν (rounded) ή να αποκοπούν (truncated) στο κατάλληλο μήκος λέξης (κατάλληλη ακρίβεια) που επιβάλλει ο υπολογιστής ή το κύκλωμα που χρησιμοποιούμε για την υλοποίηση.
- Όλα αυτά είναι προβλήματα τα οποία επηρεάζουν σημαντικά την τελική μας απόφαση σχετικά με το ποια δομή πραγματοποίησης θα πρέπει να επιλέξουμε.
- Αποδεικνύεται ότι οι διαφορετικές δομές πραγματοποίησης ενός συστήματος, ενώ είναι ισοδύναμες στην περίπτωση που χρησιμοποιούμε άπειρη ακρίβεια, παρουσιάζουν πολύ διαφορετική συμπεριφορά μεταξύ τους για πεπερασμένη ακρίβεια.



## Επίδραση Πεπερασμένου Μήκους Λέξης στην Ακρίβεια των Φίλτρων

Επίδραση της πεπερασμένης ακρίβειας συντελεστών στην υλοποίηση του φίλτρου:

Η ακρίβεια υλοποίησης των FIR και των IIR φίλτρων μειώνεται από την πεπερασμένη ακρίβεια της αναπαράστασης των συντελεστών τους, εξαιτίας:

- Της δυαδικής αναπαράστασης των αριθμών στο υπολογιστικό σύστημα που υλοποιεί το φίλτρο.
  - Της κβάντωσης των συντελεστών των φίλτρων.
  - Της στρογγυλοποίησης των ενδιάμεσων αποτελεσμάτων κατά τον υπολογισμό της εξόδου του φίλτρου (overflow errors).
- 
- Τα φαινόμενα αυτά έχουν ως αποτέλεσμα την απόκλιση των χαρακτηριστικών του ψηφιακού φίλτρου από τις επιθυμητές προδιαγραφές, οπότε το φίλτρο παύει να είναι βέλτιστο.
  - Ειδικά στην περίπτωση των φίλτρων IIR ελλοχεύει ο κίνδυνος της μετακίνησης ενός ή περισσότερων πόλων επάνω στον μοναδιαίο κύκλο ή ακόμα και εκτός αυτού, οπότε το φίλτρο παύει να είναι ευσταθές.

## Επίδραση Πεπερασμένου Μήκους Λέξης στην Ακρίβεια των Φίλτρων

Παράδειγμα :

(α) Να εξεταστεί ως προς την αιτιότητα και ευστάθεια το IIR ψηφιακό φίλτρο με συνάρτηση μεταφοράς:

$$H(z) = \frac{1 - 0.816z^{-1}}{1 - 1.927z^{-1} + 0.928176z^{-2}}$$

Απάντηση: Με παραγοντοποίηση του παρονομαστή η συνάρτηση μεταφοράς γράφεται:

$$H(z) = \frac{1 - 0.816z^{-1}}{(1 - 0.951z^{-1})(1 - 0.976z^{-1})}$$

Το IIR φίλτρο διαθέτει ένα μηδενικό στη θέση  $z = 0.816$  και δύο πραγματικούς πόλους στις θέσεις  $p_1 = 0.951$  και  $p_2 = 0.976$ . Και οι δύο πόλοι βρίσκονται εντός του μοναδιαίου κύκλου, άρα το φίλτρο είναι ευσταθές.

Για να υπολογίσουμε την κρουστική απόκριση γράφουμε τη συνάρτηση μεταφοράς σε ανάπτυγμα απλών κλασμάτων:

$$H(z) = \frac{6.4}{1 - 0.976z^{-1}} - \frac{5.4}{1 - 0.951z^{-1}}$$

Η κρουστική απόκριση είναι:

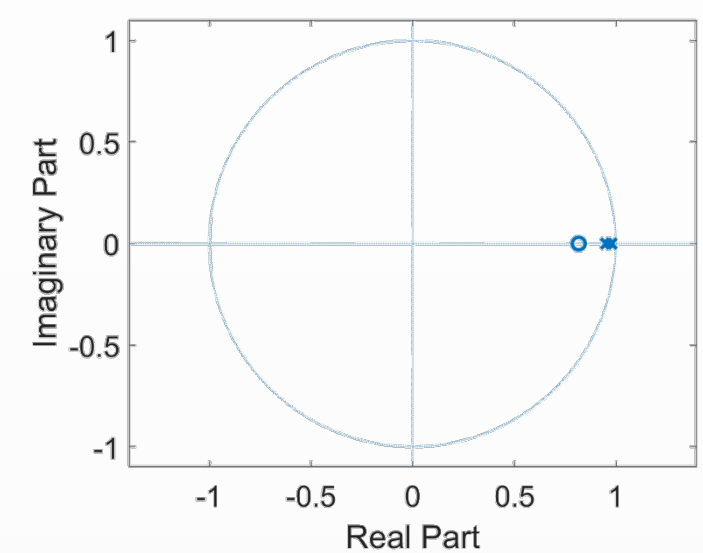
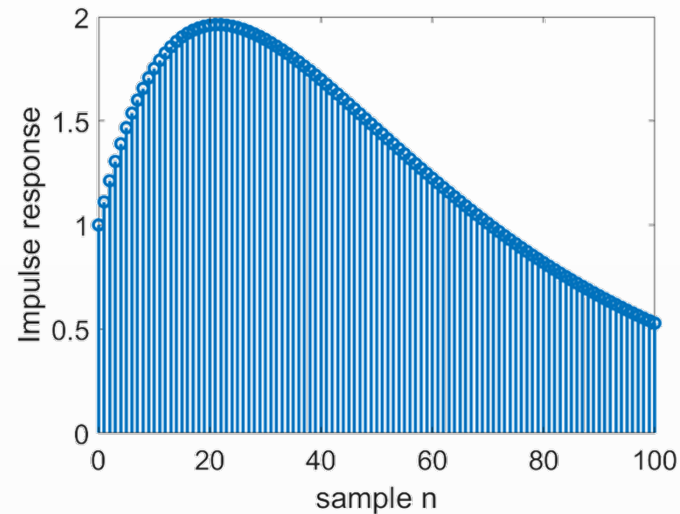
$$h[n] = [6.4 \cdot 0.976^n - 5.4 \cdot 0.951^n] u[n]$$

## Επίδραση Πεπερασμένου Μήκους Λέξης στην Ακρίβεια των Φίλτρων

Παράδειγμα (συνέχεια):

- Το σύστημα είναι αιτιατό και ευσταθές καθώς και οι δύο πόλοι είναι εντός του μοναδιαίου κύκλου

```
b=[1 -0.816]
a=[1 -1.927 0.928176]
[z,p,k] = tf2zp(b,a)
[h,n] = impz(b,a)
stem(n,h)
zplane(z,p)
```



Κρουστική απόκριση και διάγραμμα πόλων – μηδενικών

## Επίδραση Πεπερασμένου Μήκους Λέξης στην Ακρίβεια των Φίλτρων

Παράδειγμα (συνέχεια):

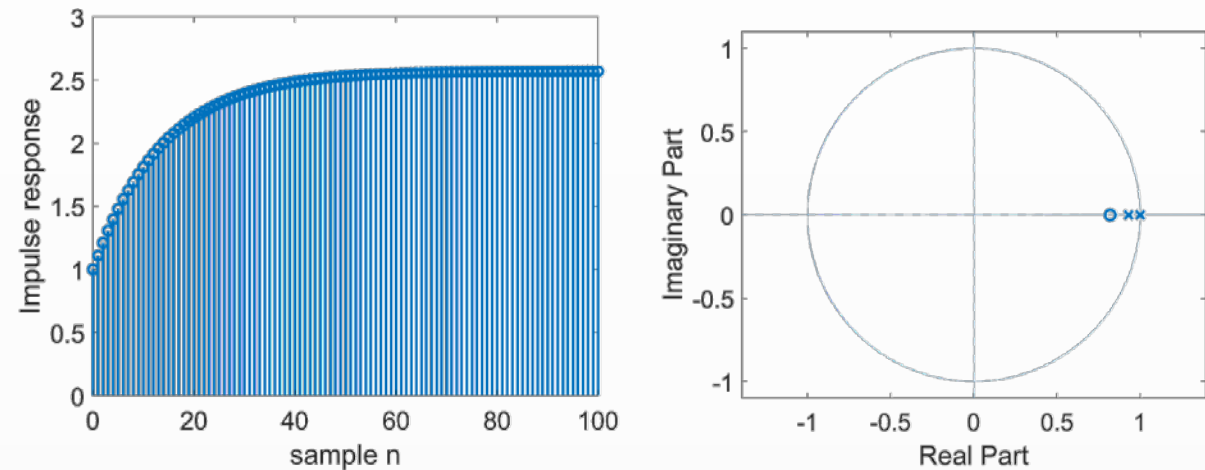
(β) Να αποτυπωθούν οι συντελεστές του φίλτρου με ακρίβεια δύο δεκαδικών ψηφίων και να επαναληφθεί η διερεύνηση.

Απάντηση: Στην περίπτωση που οι συντελεστές του φίλτρου δοθούν με **ακρίβεια δύο δεκαδικών ψηφίων**, τότε η συνάρτηση μεταφοράς γράφεται:

$$H(z) = \frac{1 - 0.82z^{-1}}{1 - 1.93z^{-1} + 0.93z^{-2}}$$

- Η κρουστική απόκριση είναι:

$$h[n] = [2.5714 \cdot 1^n - 1.5714 \cdot 0.93^n] u[n]$$



Κρουστική απόκριση και διάγραμμα πόλων – μηδενικών

- Παρατηρούμε ότι ο ένας πόλος μετακινήθηκε επάνω στον μοναδιαίο κύκλο. Επομένως το σύστημα είναι πλέον οριακά ευσταθές.

```
b=[1 -0.82]
a=[1 -1.93 0.93]
[z,p,k] = tf2zp(b,a)
[h,n] = impz(b,a)
stem(n,h)
zplane(z,p)
```

## Επίδραση Πεπερασμένου Μήκους Λέξης στην Ακρίβεια των Φίλτρων

Ειδικά για τα επαναληπτικά φίλτρα (IIR), οι μη γραμμικότητες οι οποίες οφείλονται στις πράξεις με πεπερασμένη ακρίβεια, προκαλούν συνήθως κάποιες περιοδικές ταλαντώσεις στην έξοδο, ακόμη και όταν η είσοδος είναι μηδέν ή έχει μια σταθερή τιμή.

Αυτές οι ταλαντώσεις ονομάζονται οριακοί κύκλοι (limit cycles) και οφείλονται στα σφάλματα στρογγύλευσης των γινομένων και στα σφάλματα υπερχείλισης κατά την πρόσθεση.

Το κατά πόσο καθένα από τα σφάλματα αυτά επηρεάζει την τελική συμπεριφορά ενός φίλτρου, εξαρτάται από:

- (α) το μήκος λέξης και τον τύπο της αριθμητικής που χρησιμοποιείται για την υλοποίηση,
- (β) τη μέθοδο κβάντισης των συντελεστών (λ.χ. στρογγύλευση ή αποκοπή) και
- (γ) τη δομή πραγματοποίησης του φίλτρου.

## Υλοποίηση Φίλτρων

Στα DSP chips οι βασικές λειτουργίες είναι :

- **MAC (Multiply-accumulate)**
  - 2 πράξεις κινητής υποδιαστολής : 1 πολλαπλασιασμός, 1 πρόσθεση
- **Μεταφορά δεδομένων (data shifting)**

και συνήθως υλοποιούνται με **μια** εντολή (instruction)

Έτσι το αναγκαίο πλήθος εντολών για την επεξεργασία (φιλτράρισμα) ενός δείγματος εισόδου

$$N_{instr} = 2(M + 1) + C \quad (M - \text{ταξη IIR})$$

$$N_{instr} = (M + 1) + C \quad (M - \text{ταξη FIR})$$

$C$  overhead εντολών (π.χ. Loops), τυπικά  $C \leq 10$  εξαρτάται από το chip

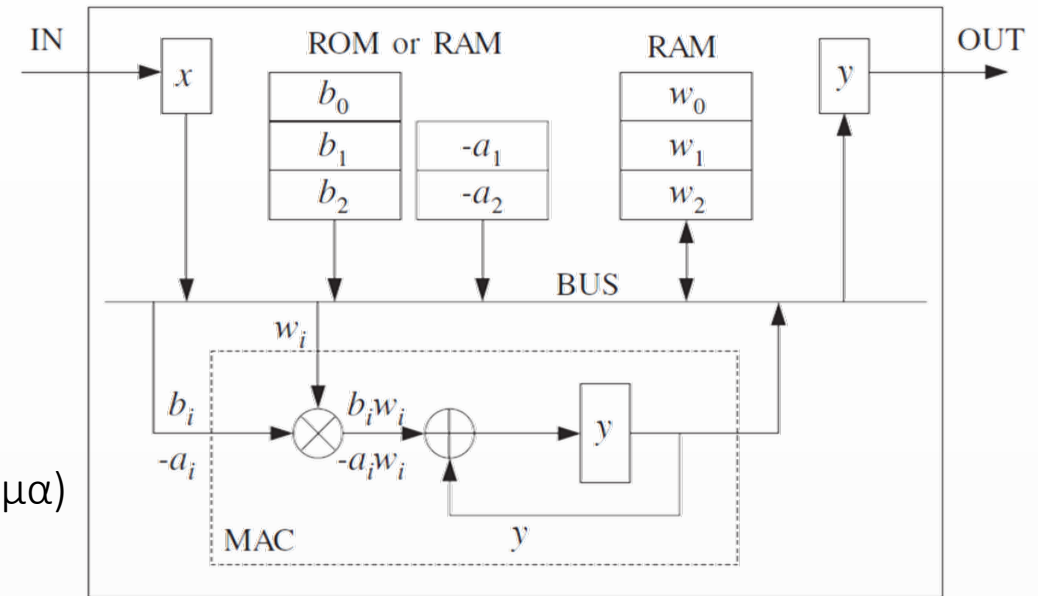
Ο συνολικός χρόνος για την επεξεργασία κάθε δείγματος εισόδου είναι :

$$T_{proc} = N_{instr} T_{instr}$$

$T_{instr}$  ο αναγκαίος χρόνος για μια εντολή MAC (π.χ. 30-80 nsec)

Ρυθμός εντολών (instruction rate)  $f_{instr} = \frac{1}{T_{instr}}$  (12.5-33.3 MIPS - million instructions per second)

Second Order Section (SOS) σε DSP



Second Order Section (SOS) αλγόριθμος

*for each input sample x do:*

$$w_0 := x - a_1 w_1$$

$$w_0 := w_0 - a_2 w_2$$

$$y := b_2 w_2$$

$$w_2 := w_1, \quad y := y + b_1 w_1$$

$$w_1 := w_0, \quad y := y + b_0 w_0$$

## Υλοποίηση Φίλτρων

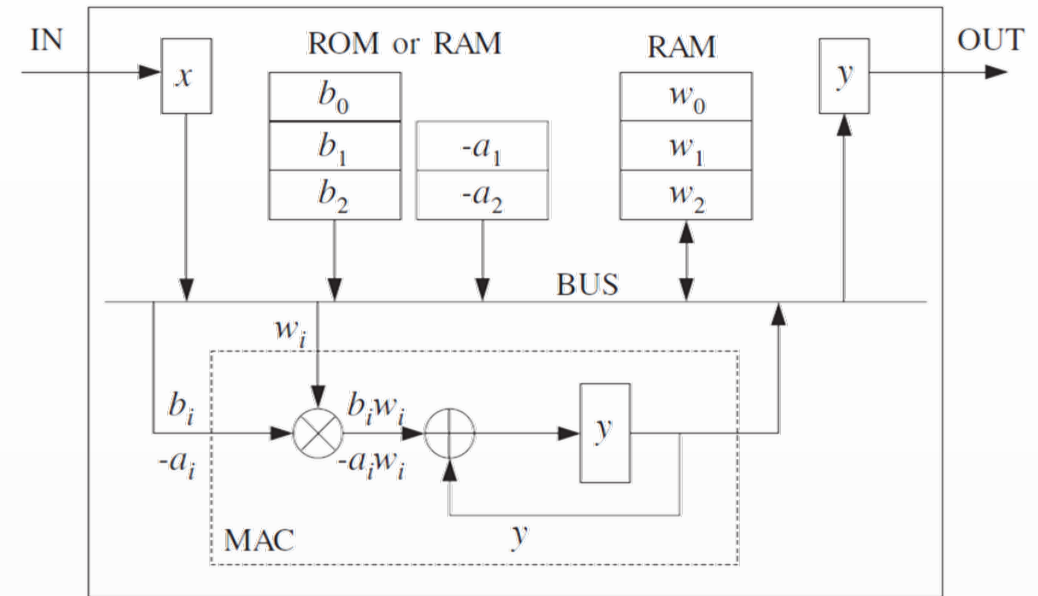
Ο χρόνος επεξεργασίας ανά δείγμα θέτει ένα άνω όριο στο ρυθμό δειγματοληψίας  $f_s$  που μπορεί να λειτουργήσει το φίλτρο

$$f_s = \frac{1}{T_{proc}} = \frac{1}{N_{instr} T_{instr}} = \frac{f_{instr}}{N_{instr}}$$

$\frac{1}{T_{proc}}$  είναι ο ρυθμός υπολογισμού ή επεξεργασίας (computational rate) του chip

- Το αναγκαίο πλήθος των εντολών ενός φίλτρου όπως και ο χρόνος ανα δείγμα εξαρτάται από τον συγκεκριμένο επεξεργαστή δηλαδή αρχιτεκτονική, σετ εντολών, προσπέλαση μνήμης, τον προγραμματισμό στην assembly του chip κ.α.

### Second Order Section (SOS) σε DSP



### Second Order Section (SOS) αλγόριθμος

Ο χρόνος επεξεργασίας ανά δείγμα θέτει ένα άνω όριο στο ρυθμό δειγματοληψίας  $f_s$  που μπορεί να λειτουργήσει το φίλτρο

$$f_s = \frac{1}{T_{proc}} = \frac{1}{N_{instr} T_{instr}} = \frac{f_{instr}}{N_{instr}}$$

$\frac{1}{T_{proc}}$  είναι ο ρυθμός υπολογισμού ή επεξεργασίας (computational rate) του chip

## Υλοποίηση Φίλτρων

Second Order Section (SOS) αλγόριθμος

- Για σύνδεση  $K$  SOS, ο συνολικός χρόνος θα είναι ίσος με το πλήθος  $K$  των SOS επί το χρόνο επεξεργασίας ενός SOS.
- Αγνοώντας επιπλέον overhead :

$$T_{SOS} \cong 5T_{instr}$$

Έτσι για τα  $K$  SOS σε σειρά ή παράλληλα (υλοποιημένα εντός του ίδιου DSP) οι συνολικές εντολές είναι

$$N_{instr} = 5K + C \text{ για } (K - \text{SOS IIR filter})$$

$C$  overhead του φίλτρου

Ο συνολικός χρόνος επεξεργασίας θα είναι

$$T_{proc} = N_{instr}T_{instr} = (5K + C)T_{instr} = KT_{SOS} + CT_{instr}$$

Αγνοώντας το μικρό όρο του overhead  $T_{proc} = KT_{SOS}$

Άρα η μέγιστη δειγματοληψίας κατά την υλοποίηση  $K$  SOS

$$f_s = \frac{1}{T_{proc}} = \frac{1}{KT_{SOS}} = \frac{f_{instr}}{5K}$$

*for each input sample  $x$  do:*

$$w_0 := x - a_1 w_1$$

$$w_0 := w_0 - a_2 w_2$$

$$y := b_2 w_2$$

$$w_2 := w_1, \quad y := y + b_1 w_1$$

$$w_1 := w_0, \quad y := y + b_0 w_0$$



## Υλοποίηση Φίλτρων

Second Order Section (SOS) αλγόριθμος

- Για παράλληλες υλοποιήσεις μπορεί να αυξηθεί ο ρυθμός throughput με τη χρήση  $K$  DSPs που λειτουργούν παράλληλα, με το καθένα να εκτελεί **μια SOS λειτουργία** σε χρόνο  $T_{SOS}$ .
- Σε αυτή την περίπτωση ο συνολικός χρόνος επεξεργασίας είναι  $T_{SOS}$  γιατί όλα τα DSP ολοκληρώνουν την επεξεργασία ταυτόχρονα και έτσι το throughput είναι  $K$  φορές γρηγορότερο από τη περίπτωση του ενός DSP
- Σε εν σειρά (cascade) υλοποιήσεις με τη χρήση  $K$  DSPs υπάρχει βελτίωση του ρυθμού throughput αλλά τα  $K$  DSPs δεν μπορούν να λειτουργήσουν ταυτόχρονα, το καθένα πρέπει να περιμένει  $T_{SOS}$  για το προηγούμενο να ολοκληρώσει
- **Pipelining** : τα  $K$  cascaded DSPs λειτουργούν ταυτόχρονα με το καθένα να επεξεργάζεται το δείγμα από την προηγούμενη χρονική στιγμή. Χρησιμοποιούνται καθυστερητές (delays) με κόστος μια συνολική καθυστέρηση στην συνολική έξοδο.

*for each input sample  $x$  do:*

$$w_0 := x - a_1 w_1$$

$$w_0 := w_0 - a_2 w_2$$

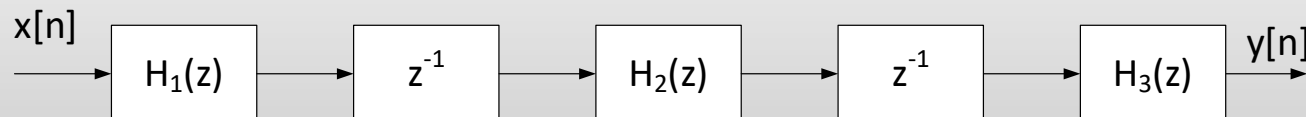
$$y := b_2 w_2$$

$$w_2 := w_1, \quad y := y + b_1 w_1$$

$$w_1 := w_0, \quad y := y + b_0 w_0$$

Για  $K$  SOS DSP's

$$H(z) = z^{-(K-1)} H_1(z) H_2(z) \dots H_K(z)$$



$$H(z) = H_1(z) z^{-1} H_2(z) z^{-1} H_3(z)$$

$$H(z) = z^{-2} H_1(z) H_2(z) H_3(z)$$

## Υλοποίηση Φίλτρων

FLOPS=floating point operations per second

MIPS=millions instructions per second

Παράδειγμα : DSP chip AT&T DSP32C

Multiply And Accumulate

Το DSP chip AT&T DSP32C κινητής υποδιαστολής εκτελεί μια βασική εντολή τύπου **MAC** σε 4 κύκλους ρολογιού (clock cycles) δηλαδή  $T_{instr} = 4T_{clock}$ . Δηλαδή ο ρυθμός εντολών (instruction rate)  $f_{instr} = f_{clock}/4$ .

MAC instruction αντιστοιχεί σε δύο πράξεις κινητής υποδιαστολής, μία πρόσθεση και ένα πολλαπλασιασμό Έτσι το chip επιτυγχάνει ρυθμό επεξεργασίας (computational rate)  $f_{FLOPS} = 2f_{instr} = f_{clock}/2$  FLOPS.

Για συχνότητα  $f_{clock} = 50\text{MHz}$  επιτυγχάνει ρυθμό εντολών  $f_{instr} = \frac{50}{4} = 12.5\text{MIPS}$   
και ρυθμό επεξεργασίας  $f_{FLOPS} = \frac{50}{2} = 25\text{MFLOPS}$  (megaflops).

Ο χρόνος ανά εντολή είναι  $T_{instr} = \frac{1}{f_{instr}} = \frac{1}{12.5} = 80\text{nsec}$

Ένα τάξης M FIR filter μπορεί να υλοποιηθεί (με κώδικα in-line) με  $N_{instr} = (M + 1) + 11 = M + 12$  (instr. per sample)

Έτσι ο χρόνος επεξεργασίας ανά sample είναι  $T_{proc} = (M + 12)T_{instr}$

## Υλοποίηση Φίλτρων

Παράδειγμα (συνέχεια) : DSP chip AT&T DSP32C

- Για ένα φίλτρο 99<sup>ης</sup> τάξης (100-tap) FIR φίλτρο στο DSP32C που λειτουργεί στα 50 MHz (clock) έχουμε

$$T_{proc} = (M + 12)T_{instr} = T_{proc} = (99 + 12)80nsec = 8.9\mu sec$$

Δηλαδή μέγιστο ρυθμό δειγματοληψίας  $f_s = \frac{1}{T_{proc}} = 112.4kHz$

## Υλοποίηση Φίλτρων

Παράδειγμα (συνέχεια) : DSP chip AT&T DSP32C

- Ένα IIR φίλτρο  $K$  τμημάτων (SOS) μπορεί να υλοποιηθεί (in-line code)

$$N_{instr} = 5K + 10 \text{ (instructions per sample)}$$

και απαιτεί ένα πλήθος wait-states κύκλων μηχανής

$$N_{wait} = 2K + 1 \text{ (wait states per sample)}$$

Συνολικά για τα  $K$  τμήματα SOS

$$T_{proc} = N_{instr} T_{instr} + N_{wait} T_{clock}$$

Όμως  $T_{instr} = 4T_{clock} = 4/f_{clock}$

$$T_{proc} = \frac{4N_{instr} + N_{wait}}{f_{clock}} = \frac{4(5K + 10) + 2K + 1}{f_{clock}}$$

- $K=1$  SOS,  $f_{clock} = 50MHz \rightarrow T_{proc} = 1.26\mu sec \rightarrow f_s = \frac{1}{T_{proc}} = 793.6kHz$
- $K=5$  SOS,  $f_{clock} = 50MHz \rightarrow T_{proc} = 3.02\mu sec \rightarrow f_s = \frac{1}{T_{proc}} = 331.1kHz$
- $K=10$  SOS,  $f_{clock} = 50MHz \rightarrow T_{proc} = 5.22\mu sec \rightarrow f_s = \frac{1}{T_{proc}} = 191.6kHz$

# Ψηφιακή Επεξεργασία Σήματος & Εικόνας

Ελληνικό Μεσογειακό Πανεπιστήμιο  
Τμήμα Ηλεκτρονικών Μηχανικών

Α. Μπακλέζος

[abaklezos@hmu.gr](mailto:abaklezos@hmu.gr)