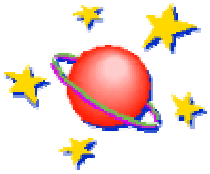


Ενότητα 5η: *Κίνδυνοι δεδομένων που απαιτούν καθυστερήσεις*

Σκοπός **Ο σκοπός της ενότητας αυτής είναι να παρουσιάσει τους κινδύνους δεδομένων που δεν αντιμετωπίζονται με την τεχνική της προώθησης. Θα ορίσουμε την έννοια κλείδωμα της σωλήνωσης και θα αναφερθούμε στην τεχνική του προγραμματισμού των εντολών. Τέλος, θα υλοποιήσουμε τον έλεγχο για τους κινδύνους δεδομένων στην ακέραια σωλήνωση, για τον υπολογιστή DLX.**

Προσδοκώμενα Αποτελέσματα **Όταν θα έχετε μελετήσει την ενότητα, θα είστε σε θέση:**

<u>να περιγράφετε το κλείδωμα της σωλήνωσης για την αποφυγή των κινδύνων δεδομένων,</u>

<u>να δημιουργείτε προγράμματα μεταγλωττιστή για μια ακολουθία εντολών, έτσι ώστε να αποφεύγονται οι καθυστερήσεις</u>,

<u>να υλοποιείτε τον έλεγχο για τη σωλήνωση στον DLX</u>.

κλείδωμα σωλήνωσης, διευθέτηση εντολής, προγραμματισμός σωλήνωσης

Κλείδωμα της σωλήνωσης

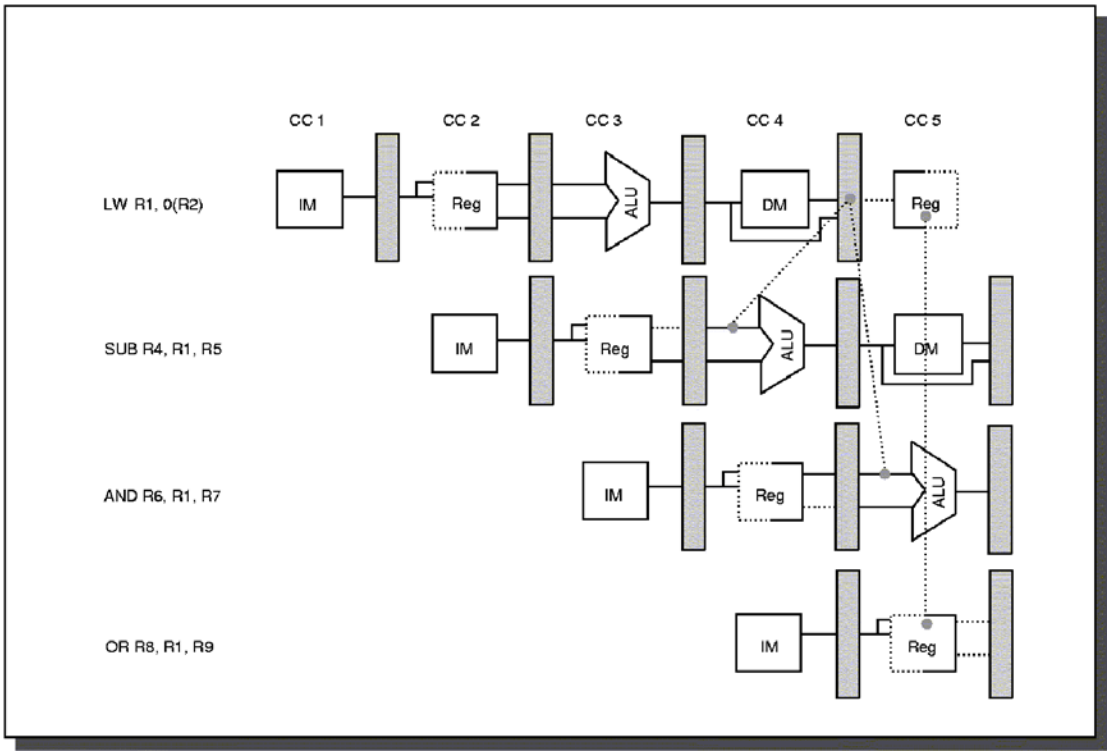
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

Έστω το παρακάτω τμήμα κώδικα ακολουθίας εντολών:

LW R1 , 0 (R2)
SUB R4 , R1 , R5
AND R6 , R1 , R7
OR R8 , R1 , R9

- **Μπορείτε να σχεδιάσετε τη δίοδο δεδομένων της σωλήνωσης με προώθηση, για την εκτέλεση των εντολών; Αν όχι, να μελετήσετε την ενότητα 4 «Κίνδυνοι δεδομένων».**
- **Να προσδιορίσετε τα σημεία στα οποία μπορεί να παρουσιαστεί κίνδυνος δεδομένων.**
- **Θεωρείτε πως οι κίνδυνοι που προκύπτουν μπορούν να αντιμετωπιστούν με την τεχνική της προώθησης;**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 1



ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 1

Στο σχήμα απεικονίζεται η δίοδος δεδομένων της σωλήνωσης με προώθηση. Η εντολή LW δεν έχει δεδομένα μέχρι το τέλος του 4ου κύκλου ρολογιού (δηλαδή τον κύκλο MEM), ενώ η εντολή SUB χρειάζεται δεδομένα από την αρχή αυτού του κύκλου ρολογιού. Στην περίπτωση αυτή, ο κίνδυνος δεδομένων που προέρχεται από τη χρήση του αποτελέσματος μίας εντολής φόρτωσης δεν μπορεί να απαλειφθεί πλήρως με απλό υλικό. Στο σχήμα φαίνεται ότι ένα τέτοιο μονοπάτι προώθησης θα έπρεπε να λειτουργεί ανάστροφα στο χρόνο. Κάτι τέτοιο όμως δεν είναι ακόμα εφικτό. Μπορούμε να προχωρήσουμε το αποτέλεσμα κατευθείαν στην ALU για την εντολή AND, η οποία αρχίζει 2 κύκλους ρολογιού μετά τη φόρτωση. Η εντολή OR δεν αντιμετωπίζει κανένα πρόβλημα γιατί δέχεται την τιμή μέσω του αρχείου καταχωρητών.
Συμπερασματικά μπορούμε να πούμε πως εντολή φόρτωσης μπορεί να προωθήσει τα αποτελέσματα της στις εντολές AND και OR, αλλά όχι στη SUB, αφού κάτι τέτοιο θα σήμαινε προώθηση του αποτελέσματος σε αρνητικό χρόνο.

Η προώθηση δεν μπορεί να αντιμετωπίσει μία καθυστέρηση της εντολής φόρτωσης, όπως φαίνεται στη δραστηριότητα 1. Για να αντιμετωπίσουμε το πρόβλημα αυτό πρέπει να προσθέσουμε κατάλληλο υλικό, το οποίο ονομάζεται **κλείδωμα σωλήνωσης**.

Το κλείδωμα της σωλήνωσης ανιχνεύει έναν κίνδυνο και καθυστερεί τη σωλήνωση μέχρι να απαλειφθεί ο κίνδυνος αυτός. Έτσι, το κλείδωμα της σωλήνωσης καθυστερεί τη σωλήνωση ξεκινώντας με την εντολή εκείνη, η οποία θέλει να χρησιμοποιήσει τα αποτελέσματα, τα οποία όμως δεν έχει ακόμα παράγει η πηγαία εντολή. Μάλιστα, το κλείδωμα καθυστερεί τη σωλήνωση αρκετά ώστε η πηγαία εντολή να έχει παράγει τα αποτελέσματα.

Παράδειγμα

Υποθέστε ότι το 30% των εντολών είναι εντολές φόρτωσης, και τις μισές φορές, η εντολή που ακολουθεί μία εντολή φόρτωσης εξαρτάται από το αποτέλεσμα της φόρτωσης. Αν αυτός ο κίνδυνος προκαλεί καθυστέρηση ενός κύκλου, πόσο ταχύτερος είναι ο ιδανικά σωληνωμένος υπολογιστής (με CPI= 1), ο οποίος δεν καθυστερεί τη σωλήνωση, από την πραγματική σωλήνωση; Αγνοείστε όποιες καθυστερήσεις δεν είναι καθυστερήσεις σωλήνωσης.

Απάντηση :

Το CPI για μια εντολή που ακολουθεί μια εντολή φόρτωσης είναι 1.5, αφού προκαλεί καθυστέρηση τις μισές φορές. Επειδή οι εντολές φόρτωσης αποτελούν το 30% του συνόλου, το πραγματικό CPI είναι:

(Εντολές προγράμματος–Εντολές φόρτωσης)*$CPI_{ιδανικό}$ + (Εντολές φόρτωσης)*$CPI_{1,5}$ = (1-0.3)*1+0.3*1.5 =1.15

Ο ιδανικός υπολογιστής θα είναι ταχύτερος κατά τον λόγο των CPIs.
Άρα ο ιδανικός υπολογιστής είναι 1.15 φορές ταχύτερος.

Προγραμματισμός μεταγλωττιστή για κίνδυνους δεδομένων

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

- **Θυμάστε τον τρόπο που δημιουργείται ο κώδικας για την εκτέλεση της πράξης:**
  A = B + C
  όπου A, B, C είναι διευθύνσεις μνήμης; Για περισσότερες λεπτομέρειες καλό θα ήταν να ανατρέξετε στις σημειώσεις «Αρχιτεκτονική Υπολογιστών Ι» και συγκεκριμένα στο 2^{ο} κεφάλαιο: «Αρχιτεκτονικές Συνόλου Εντολών».
- **Να γράψετε την ακολουθία εντολών η οποία εκτελεί την παραπάνω πράξη.**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 2

Για την εκτέλεση της πράξης:
A = B + C
θα χρειαστούμε 3 καταχωρητές. Έστω οι καταχωρητές R1, R2, R3 οι οποίοι θα αποθηκεύσουν τις διευθύνσεις μνήμης B, C και A αντίστοιχα. Αρχικά πρέπει να φορτώσουμε τις διευθύνσεις B, C στους καταχωρητές R1, R2 αντίστοιχα και στη συνέχεια να προσθέσουμε τα περιεχόμενα των δύο καταχωρητών. Το αποτέλεσμα της πρόσθεσης θα αποθηκευτεί στον καταχωρητή R3 και στη συνέχεια θα γίνει εγγραφή αυτού στη διεύθυνση μνήμης A.
Η ακολουθία των εντολών είναι:
LW R1 , B
LW R2 , C
ADD R3 , R1 , R2
SW A , R3

Το τυπικό σχέδιο δημιουργίας ενός κώδικα για μία πράξη όπως A = B + C όπου A, B, C είναι διευθύνσεις στη μνήμη, παράγει καθυστέρηση για μία φόρτωση της δεύτερης τιμής δεδομένων της C. Στο σχήμα 1.5.1 παρουσιάζεται η ακολουθία εντολών στον υπολογιστή DLX, για την πράξη της πρόσθεσης.

LW R1 , B	IF	ID	EX	MEM	WB				
LW R2 , C		IF	ID	EX	MEM	WB			
ADD R3 , R1 , R2			IF	ID	Stall	EX	MEM	WB	
SW A , R3				IF	Stall	ID	EX	MEM	WB

Σχήμα 1.5.1 - Η ακολουθία κώδικα DLX υπολογιστή για την εντολή A = B + C . Η ADD εντολή πρέπει να καθυστερήσει έτσι ώστε να επιτρέψει στη φόρτωση της C να ολοκληρωθεί. Η SW δεν χρειάζεται να καθυστερήσει άλλο, επειδή το υλικό προώθησης περνά το αποτέλεσμα από τον καταχωρητή MEM/WB απευθείας στην είσοδο της μνήμης δεδομένων για αποθήκευση.

Ο μεταγλωττιστής σε έναν σωληνωμένο υπολογιστή προσπαθεί να προγραμματίσει τη σωλήνωση, έτσι ώστε να αποφύγει τις καθυστερήσεις, αλλάζοντας τη σειρά εκτέλεσης των εντολών για να εξαλείψει των κίνδυνο δεδομένων. Για παράδειγμα ο μεταγλωττιστής μπορεί να δοκιμάσει να αποφύγει να παράγει ένα κώδικα με μια εντολή φόρτωσης, η οποία θα ακολουθείται από την άμεση χρήση του καταχωρητή προορισμού της εντολής φόρτωσης. Η τεχνική αυτή ονομάζεται **προγραμματισμός σωλήνωσης** ή **προγραμματισμός εντολών**.

Παράδειγμα

Δημιουργήστε DLX κώδικα που αποφεύγει τις καθυστερήσεις σωλήνωσης για την παρακάτω ακολουθία:

a = b + c;
d = e – f ;

όπου a, b, c, d, e, f είναι διευθύνσεις μνήμης. Υποθέστε ότι οι φορτώσεις έχουν αδράνεια ενός κύκλου ρολογιού.

Απάντηση:

Ο κώδικας για τη δεδομένη ακολουθία εντολών , ώστε να αποφεύγονται οι καθυστερήσεις είναι:

```
LW   Rb, b
LW   Rc, c
LW   Re, e          ; ανταλλαγή της θέσης των εντολών για αποφυγή της καθυστέρησης
ADD Ra, Rb, Rc
LW   Rf, f
SW a, Ra            ; ανταλλαγή αποθήκευσης /φόρτωσης για να αποφύγουμε καθυστερήσεις
SUB Rb, Re, Rf
SW   d, Rd
```

Παρατηρήστε ότι η χρήση διαφορετικών καταχωρητών για την πρώτη και δεύτερη δήλωση είναι απαραίτητη, για να είναι σωστός ο προγραμματισμός. Αν η μεταβλητή e φορτωνόταν στον ίδιο καταχωρητή με τη μεταβλητή b ή τη μεταβλητή c, ο προγραμματισμός δε θα ήταν σωστός.

Υλοποίηση του ελέγχου για τη σωλήνωση στον DLX

Η διαδικασία, η οποία επιτρέπει σε μία εντολή να κινηθεί από το στάδιο αποκωδικοποίησής (ID) στο στάδιο εκτέλεσης (EX), ονομάζεται **διευθέτηση εντολής**. Μία εντολή που έκανε αυτό το βήμα λέμε ότι διευθετήθηκε.

Στο σχήμα 1.5.2 συνοψίζουμε όλες τις περιπτώσεις κινδύνων δεδομένων που μπορεί να συναντήσουμε στη σωλήνωση.

Κατάσταση	**Παράδειγμα κώδικα**	**Ενέργεια**

Ανεξαρτησία	LW **R1**, 45 (R2)
	ADD R5, R6, R7
	SUB R8, R6, R7
	OR R9, R6, R7
		Δεν συμβαίνει κανένας κίνδυνος δεδομένων, αφού δεν υπάρχει εξάρτηση από τον R1 στις αμέσως επόμενες τρεις εντολές.

Εξάρτηση που απαιτεί καθυστέρηση (stall)	LW **R1**, 45 (R2)
	ADD R5, **R1**, R7
	SUB R8, R6, R7
	OR R9, R6, R7
		Ο συγκριτής ανιχνεύει τη χρήση του R1 στην εντολή ADD και καθυστερεί την ADD (και τις SUB και OR) πριν η ADD αρχίσει το στάδιο EX.

Εξάρτηση που αντιμετωπίζεται με προώθηση	LW **R1**, 45 (R2)
	ADD R5, R6, R7
	SUB R8, **R1**, R7
	OR R9, R6, R7
		Ο συγκριτής ανιχνεύει τη χρήση του R1 στην εντολή SUB και προωθεί το αποτέλεσμα της φόρτωσης στην ALU τη στιγμή που για την SUB ξεκινά το EX.

Εξάρτηση που αντιμετωπίζεται με πρόσβαση σε σειρά	LW **R1**, 45 (R2)
	ADD R5, R6, R7
	SUB R8, R6, R7
	OR R9, **R1**, R7
		Καμία πράξη δεν απαιτείται γιατί η ανάγνωση του R1 στην OR γίνεται στο 2° μισό του σταδίου ID, ενώ η εγγραφή της OR γίνεται στο 1° μισό του σταδίου WB.

Σχήμα 1.5.2 - Περιπτώσεις που το υλικό ανίχνευσης κινδύνων της σωλήνωσης μπορεί να αντιμετωπίσει, συγκρίνοντας τους προορισμούς και τις πηγές διαδοχικών εντολών. Αυτός ο πίνακας δείχνει πως η μόνη σύγκριση που χρειάζεται είναι μεταξύ του τελεστέου προορισμού μιας εντολής και των πηγαίων τελεστέων των δύο εντολών που έπονται. Στην περίπτωση μίας καθυστέρησης, οι εξαρτήσεις της σωλήνωσης θα μοιάζουν με την τρίτη περίπτωση, μόλις συνεχιστεί η εκτέλεση.

Όταν ένας κίνδυνος ανιχνευθεί τότε:

- Η μονάδα ελέγχου πρέπει να εισάγει την καθυστέρηση της σωλήνωσης και να εμποδίσει τις εντολές που βρίσκονται στο IF ή στο στάδιο ID να προχωρήσουν.

- Επειδή όλες οι πληροφορίες ελέγχου μεταφέρονται στους καταχωρητές σωλήνωσης, αρκεί όταν ανιχνεύσουμε ένα κίνδυνο, να αλλάξουμε το τμήμα ελέγχου του ID/EX καταχωρητή σωλήνωσης σε μηδενικά, ώστε να έχουμε μια no-operation εντολή.

- Το περιεχόμενο του καταχωρητή IF/ID φυλάσσεται, ώστε να κρατήσουμε την καθυστερούμενη εντολή.

Η εφαρμογή της λογικής της προώθησης είναι παρόμοια, παρότι υπάρχουν περισσότερες περιπτώσεις που πρέπει να λάβουμε υπόψη μας.
Όλη η προώθηση γίνεται από:

την έξοδο της ALU	→	είσοδο της ALU
ή	**στην**	ή
την έξοδο της μνήμης δεδομένων	→	είσοδο της μνήμης δεδομένων
		ή
		μονάδα ανίχνευσης του 0

Έτσι μπορούμε να εφαρμόσουμε την προώθηση με μια σύγκριση των καταχωρητών προορισμού του IR που περιέχονται στα EX/MEM και MEM/WB στάδια, ως προς τους πηγαίους καταχωρητές του IR που περιέχονται στους ID/EX και EX/MEM καταχωρητές.

Από την άλλη μεριά, με τους συγκριτές και τη συνδυαστική λογική που χρειαζόμαστε για να καθορίσουμε πότε ένα μονοπάτι προώθησης πρέπει να ενεργοποιηθεί, είναι αναγκαίο:

- να μεγαλώσουμε τους πολυπλέκτες στις εισόδους της ALU
- να προσθέσουμε τις συνδέσεις από τους καταχωρητές σωλήνωσης που χρησιμοποιούνται για προώθηση των αποτελεσμάτων.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Να περιγράψετε συνοπτικά τις λειτουργίες που ακολουθούν την ανίχνευση ενός κινδύνου, ο οποίος δεν αντιμετωπίζεται με την τεχνική της προώθησης.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Έστω ο κώδικας ο οποίος πρόκειται να υλοποιηθεί με την τεχνική της σωλήνωσης:

```
LW   R24, 0(R16)
LW   R25, 4(R16)
SLT  R8,  R25, R24
```

- **Να περιγράψετε την εκτέλεση των εντολών.**
- **Να εντοπίσετε που προκύπτουν κίνδυνοι δεδομένων.**
- **Να προτείνεται τρόπους αντιμετώπισης των κινδύνων δεδομένων και να αιτιολογήσετε την απάντησή σας.**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4

Οι κίνδυνοι δεδομένων που προκύπτουν είναι οι ακόλουθοι:

1ος: Ο πρώτος πηγαίος τελεστέος της εντολής SLT εξαρτάται από το αποτέλεσμα της δεύτερης εντολής LW.
2ος: Ο δεύτερος πηγαίος τελεστέος της εντολής SLT εξαρτάται από το αποτέλεσμα της πρώτης εντολής LW.

Για την αντιμετώπιση του 1ου κινδύνου δεν μπορούμε να υλοποιήσουμε την τεχνική της προώθησης γιατί θα έπρεπε να έχουμε προώθηση σε αρνητικό χρόνο. Επίσης ο μεταγλωττιστής δεν μπορεί να προγραμματίσει τη σωλήνωση γιατί οι εντολές αλληλεξαρτώνται. Επομένως θα χρησιμοποιήσουμε το κλείδωμα της σωλήνωσης.

Έτσι προκύπτει η εξής ακολουθία κώδικα:

	1ος	2ος	3ος	4ος	5ος	6ος	7ος	8ος
LW R24, 0(16)	IF	ID	EX	MEM	WB			
LW R25, 4(16)		IF	ID	EX	MEM	WB		
SLT R8, R25, R24			IF	ID	stall	EX	MEM	WB

Η πρώτη εντολή LW παίρνει την τιμή του καταχωρητή R24 από το πεδίο .MDR του καταχωρητή σωλήνωσης MEM/WB στο τέλος του 4ου κύκλου ρολογιού.
Η δεύτερη εντολή LW παίρνει την τιμή του καταχωρητή R25, από το πεδίο .MDR του καταχωρητή σωλήνωσης MEM/WB, στο τέλος του 5ου κύκλου ρολογιού.
Η εντολή SLT χρειάζεται την τιμή του καταχωρητή R25 στον 5ο κύκλο ρολογιού, στην αρχή του σταδίου EX. 'Ομως το αποτέλεσμα, δεν μπορεί να προωθηθεί από τον καταχωρητή σωλήνωσης MEM/WB της 2ης εντολής LW προς τα πίσω, στο στάδιο EX της εντολής SLT. Γι' αυτό η εντολή SLT πρέπει να καθυστερήσει, προκειμένου πάρει τη σωστή τιμή. Γι΄ αυτό κλειδώνουμε τη σωλήνωση για έναν κύκλο ρολογιού.

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4

Στον 6^{ο} κύκλο ρολογιού, η εντολή SLT βρίσκεται στο στάδιο EX.
Σε αυτό το στάδιο, γενικά για όλες τις εντολές, στην ALU δημιουργείται η διεύθυνση του τελεστέου, η οποία τοποθετείται στο πεδίο .ALUoutput, του καταχωρητή σωλήνωσης EX/MEM δηλαδή στο πεδίο EX/MEM.ALUoutput. Στο στάδιο MEM διαβάζεται η τιμή του τελεστέου από τη διεύθυνση, η οποία(διεύθυνση) βρίσκεται στο πεδίο MEM/WB.MDR
Συνεπώς, η εντολή SLT παίρνει την τιμή του καταχωρητή R25 από το αρχείο καταχωρητών στο δεύτερο μισό του 6ου κύκλου ρολογιού, όπου έχει προλάβει και το έχει γράψει στο πρώτο μισό του ίδιου κύκλου η δεύτερη εντολή LW.
Η εντολή SLT παίρνει την τιμή του καταχωρητή R24, για την κάτω είσοδο της ALU, από το αρχείο καταχωρητών, όπου την έχει γράψει η πρώτη εντολή LW από τον προηγούμενο κύκλο ρολογιού.

Ανακεφαλαιώνοντας λοιπόν …

Τους κίνδυνους δεδομένων, τους οποίους δεν μπορούμε να αντιμετωπίσουμε με προώθηση, τους αντιμετωπίζουμε με το κλείδωμα σωλήνωσης και με την τεχνική του προγραμματισμού σωλήνωσης.

Στο κλείδωμα σωλήνωσης προσθέτουμε νέο υλικό, το οποίο ανιχνεύει τον κίνδυνο δεδομένων και καθυστερεί τη σωλήνωση.

Με την τεχνική του προγραμματισμού σωλήνωσης ο μεταγλωττιστής προγραμματίζει τη σωλήνωση έτσι ώστε ο κώδικας να τακτοποιείται με τέτοιον τρόπο που να μην έχουμε εξαρτήσεις δεδομένων μεταξύ των εντολών, οι οποίες προκαλούν καθυστερήσεις στη σωλήνωση.