Αρχιτεκτονική Υπολογιστών

Ενότητα 6: Διασωλήνωση

Δρ. Μηνάς Δασυγένης
mdasyg@ieee.org
Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών
http://arch.ece.uowm.gr/mdasyg

BY NC ND
Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης
ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

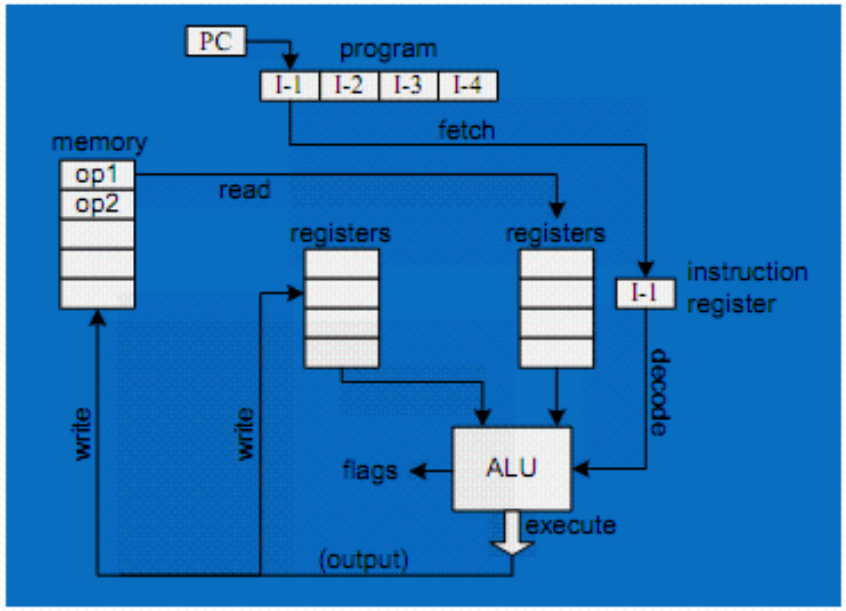
Σκοπός της ενότητας

- Η βελτίωση των επιδόσεων μέσω της τεχνικής της διασωλήνωσης.
- Τα προβλήματα που δημιουργούνται από τη διασωλήνωση.

Διασωλήνωση

Ο επεξεργαστής εκτελεί συνεχώς 5 λειτουργίες

- Fetch.
- Decode.
- Fetch operands.
- Execute.
- Store output.



Διασωλήνωση με απλά λόγια

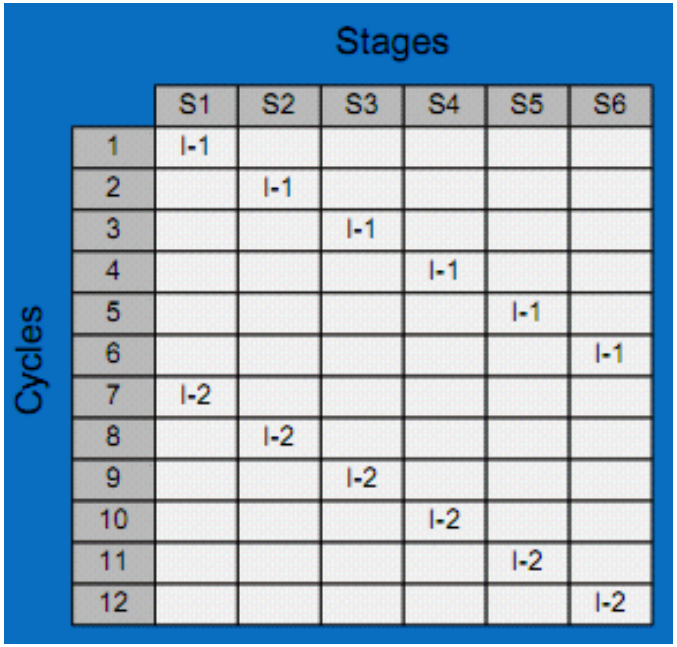
- Δημοφιλής Τεχνική *(χρησιμοποιείται παντού).*
- Η εκτέλεση πολλών εντολών γίνεται σε επικάλυψη *(~παράλληλα).*
- Στην x86 υπάρχουν μόνο 2 βαθμίδες ή στάδια *(CPU και BIU {bus interface unit}).*
- Σε νεότερους επεξεργαστές υπάρχουν περισσότερα στάδια *(κάτω από 20).*

Διασωλήνωση (1/3)

- Η διασωλήνωση κάνει τον επεξεργαστή να εκτελεί **τις εντολές παράλληλα.**
- Ο μεγάλος αριθμός των εντολών εκτελείται γρήγορα.
- Η εκτέλεση εντολής διαιρείται σε **διακριτά τμήματα.**
- Κάθε στάδιο είναι **αυτόνομο**, και απομονώνεται από τα γειτονικά στάδια.
- Η χρήση του επεξεργαστή γίνεται πιο αποδοτική.
- Υψηλότερες συχνότητες του συστήματος και περισσότερα στάδια διασωλήνωσης συνδυάζονται για την αύξηση της απόδοσης του επεξεργαστή.

Ένας επεξεργαστής χωρίς διασωλήνωση δεν είναι αποδοτικός

Για την εκτέλεση **n** εντολών και μέσο χρόνο εκτέλεσης κ, απαιτούνται **n*k** μονάδες χρόνου.



Έστω έχουμε k=6 στάδια για την πλήρη εκτέλεση
μιας εντολής:

- Fetch,
- Decode,
- fetch operands,
- Execute,
- write memory,
- write registerfile.

Για την εκτέλεση n=2 εντολών τότε θα έχουμε n*k = 12 κύκλους.

Υποχρεωτικά: Όλες οι εντολές διέρχονται από όλα τα στάδια.
Υπόθεση: Όλα τα στάδια μπορούν να εκτελούνται παράλληλα.

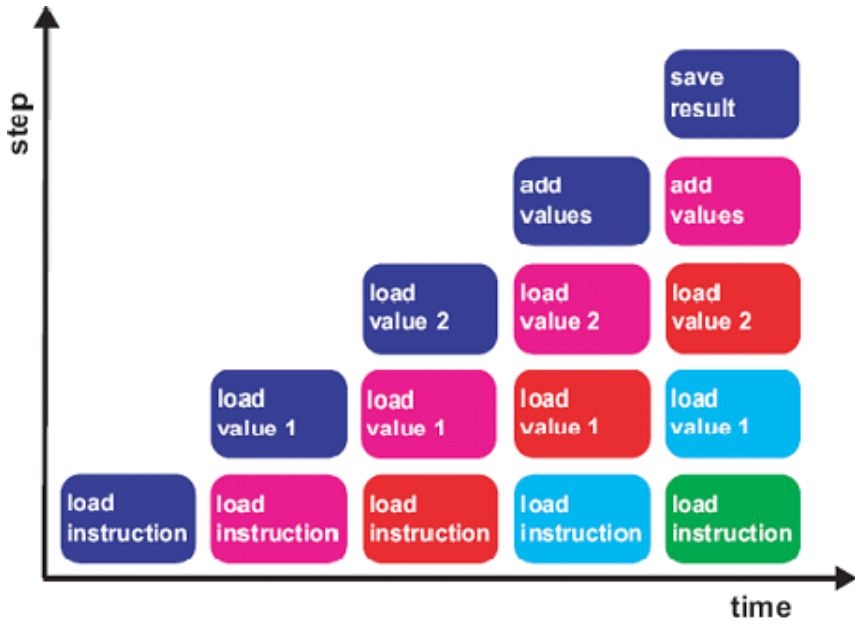
Παράδειγμα υπολογιστή χωρίς διασωλήνωση

Παράδειγμα:

- Για μια πρόσθεση 2 αριθμών σε μια CPU με 5 στάδια ανά εντολή απαιτούνται οι εξής ενέργειες *(απαιτούνται **5*n** κύκλοι):*
- Φόρτωση εντολής πρόσθεσης στο CPU.
- Φόρτωση δεδομένων πρόσθεσης αθροιστή A σε ένα καταχωρητή CPU.
- Φόρτωση δεδομένων πρόσθεσης αθροιστή B σε ένα καταχωρητή CPU.
- Εκτέλεση εντολής.
- Μεταφορά αποτελέσματος στη μνήμη.

Βελτίωση του χρόνου εκτέλεσης με διασωλήνωση

- Όλες οι λειτουργικές μονάδες χρησιμοποιούνται ταυτόχρονα, ώστε σε κάθε κύκλο να παράγεται αποτέλεσμα.
- Η συνολικός χρόνος βελτιώνεται και γίνεται (n-1)+5.
- Χωρίς διασωλήνωση ο χρόνος n*5.



Διασωλήνωση (2/3)

- Διασωλήνωση είναι η σύνδεση σε σειρά επεξεργαστικών στοιχείων, έτσι ώστε η έξοδος ενός στοιχείου να είναι η είσοδος στο επόμενο.
- Η ονομασία έρχεται από το **σωλήνα του νερού**. Νερό εισέρχεται συνεχώς στο σωλήνα, χωρίς να χρειάζεται να περιμένουμε να βγει από την άλλη άκρη.
- Οδηγεί στη μείωση του κρίσιμου μονοπατιού.
- Υπάρχουν στάδια επεξεργασίας. Κάθε στάδιο ολοκληρώνει ένα κομμάτι της εντολής του επεξεργαστή.

Η διασωλήνωση οδηγεί στη μείωση του κρίσιμου μονοπατιού

Η ταχύτητα λειτουργίας ενός ψηφιακού συστήματος ή η περίοδος λειτουργίας του ρολογιού εξαρτάται από το μεγαλύτερο μονοπάτι:

- το πιο μεγάλο μονοπάτι ανάμεσα σε 2 μανδαλωτές *(latches)*, ή
- το πιο μεγάλο μονοπάτι ανάμεσα σε μια είσοδο και ένα μανδαλωτή, ή
- το πιο μεγάλο μονοπάτι από ένα μανδαλωτή και την έξοδο, ή
- το πιο μεγάλο μονοπάτι από την είσοδο προς την έξοδο.

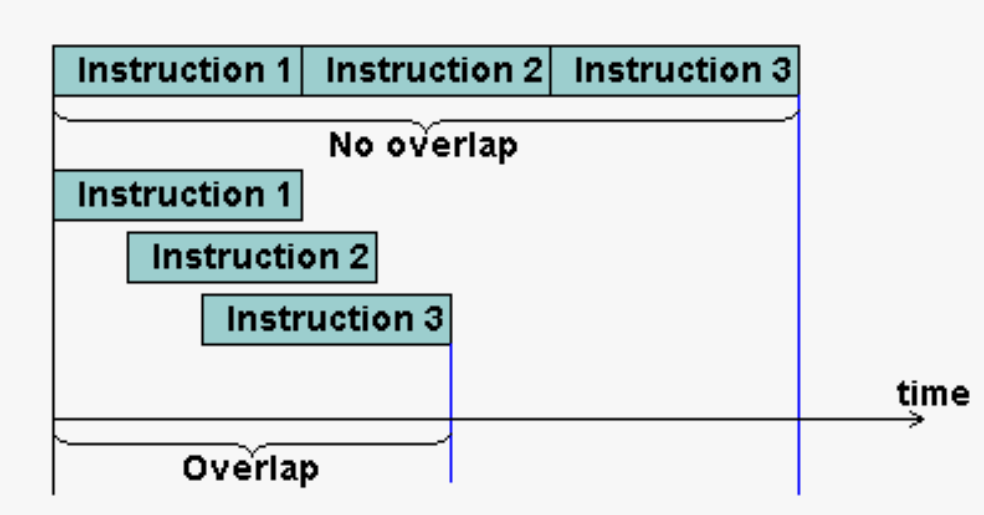
Η πιο μεγάλη καθυστέρηση, ορίζει τη μέγιστη συχνότητα λειτουργίας.

Αν χρησιμοποιηθεί μεγαλύτερη συχνότητα λειτουργίας, τότε θα δυσλειτουργεί το κύκλωμα, αφού δε θα είναι δυνατό να μεταδοθεί το σήμα στο συγκεκριμένο μονοπάτι, μέσα στο συγκεκριμένο χρονικό διάστημα.

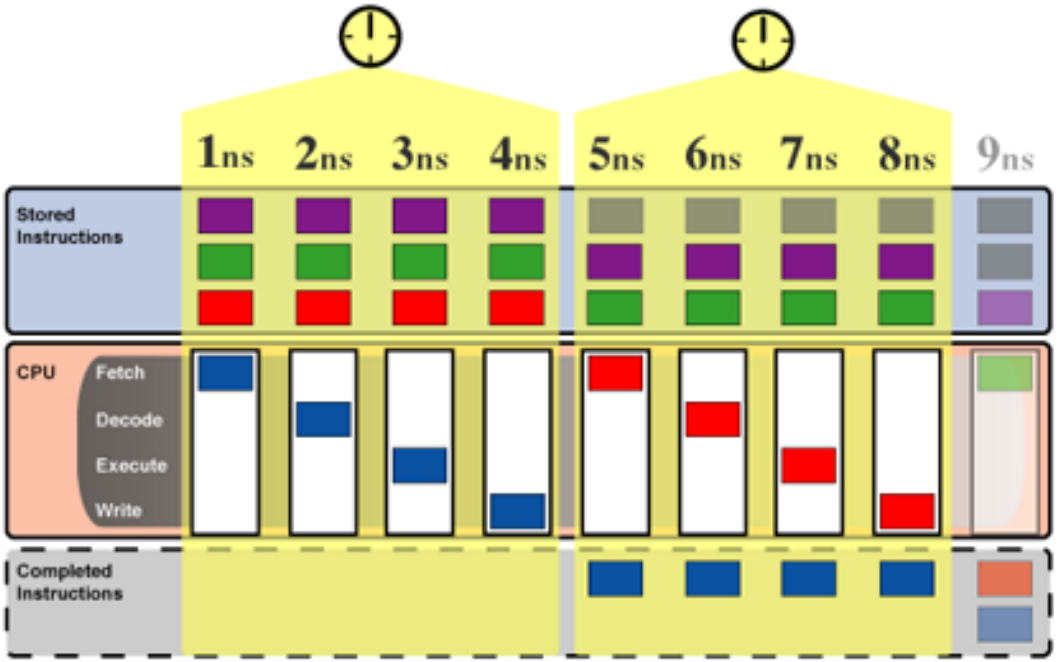
Η στρατηγική τοποθέτηση μανδαλωτών, μειώνει το κρίσιμο μονοπάτι.

Διασωλήνωση (3/3)

- Η διασωλήνωση ΔΕΝ αυξάνει την ταχύτητα εκτέλεσης μιας εντολής.
- Η διασωλήνωση αυξάνει το ρυθμό απόδοσης *(αριθμό εκτελούμενων εντολών στη μονάδα του χρόνου)*, με την επικάλυψη εκτέλεσης εντολών.

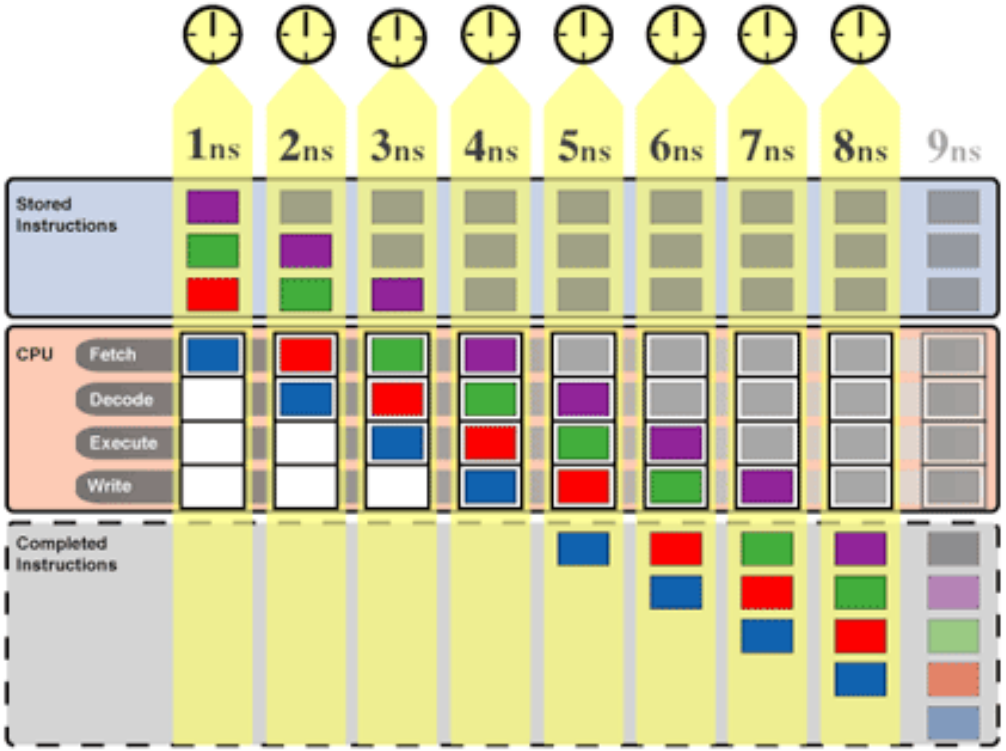


Ένας υπολογιστής 4 σταδίων εκτέλεσης χωρίς διασωλήνωση



Σε 9 ns έχουν ολοκληρωθεί 2 εντολές.

Ένας υπολογιστής με διασωλήνωση



Σε 9 ns έχουν ολοκληρωθεί 5 εντολές.

Στάδια διασωλήνωσης σε επεξεργαστές

- Ο Pentium IV είχε 20 επίπεδα διασωλήνωσης.
- Ο Prescott Pentium 30 επίπεδα διασωλήνωσης *(hyper-pipelining).*
  - Πολύ καλή απόδοση, αλλά και πολλά προβλήματα.
- Intel Core i7 *(Nehalem)* 14 επίπεδα διασωλήνωσης, εκτέλεση έως 4 εντολές ανά κύκλο *(IPC – Instructions per clock).*
- Τα πολλά στάδια δημιουργούν πολλά προβλήματα.

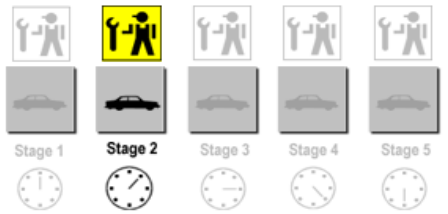
Διασωλήνωση: Που βασίζεται

- Οι επεξεργαστές λειτουργούν συνεχώς ως εξής:
  - **Fetch Instruction (A).**
    - ✓ Store Instruction in Register.
  - **Decode Instruction (B).**
    - ✓ Increment the address on program counter.
  - **Execute the Instruction (C).**
  - **Write the results (D).**

➔ **4** είναι τα πιο απλά στάδια της διασωλήνωσης.

Διασωλήνωση παράδειγμα στην αυτοκινητοβιομηχανία (1/3)

- **Στάδιο 1:** Κατασκευή του πλαισίου του οχήματος *(σασί).*
- **Στάδιο 2:** Τοποθέτηση του κινητήρα στο σκελετό του οχήματος.
- **Στάδιο 3:** Τοποθέτηση των περιφερειακών κομματιών του αμαξώματος *(πόρτες, κάλυμμα, επένδυση)* στο πλαίσιο του οχήματος.
- **Στάδιο 4:** Τοποθέτηση των τροχών.
- **Στάδιο 5:** βάψιμο του SUV.



Διασωλήνωση παράδειγμα στην αυτοκινητοβιομηχανία (2/3)

Μπορούμε να δημιουργήσουμε και άλλα επίπεδα διασωλήνωσης στο παράδειγμα της αυτοκινητοβιομηχανίας:

- **Στάδιο 1:** Κατασκευή του πλαισίου του οχήματος:
  - *Πλήρωμα 1a:* Τοποθέτηση των εξαρτημάτων μεταξύ τους και συγκόλληση επιτόπου για να ενωθούν.
  - *Πλήρωμα 1b:* Πλήρης συγκόλληση με όλα τα κομμάτια του αυτοκινήτου.
- **Στάδιο 2:** Τοποθέτηση του κινητήρα στο σκελετό του οχήματος:
  - *Πλήρωμα 2a:* Τοποθέτηση της μηχανής στο σκελετό του αυτοκινήτου και προσάρμοση την στην σωστή θέση.
  - *Πλήρωμα 2b:* Σύνδεση μηχανής με τα στοιχεία μετάδοσης αυτοκινήτου.

Διασωλήνωση παράδειγμα στην αυτοκινητοβιομηχανία (3/3)

- **Στάδιο 3:** Τοποθέτηση των περιφερειακών κομματιών του αμαξώματος *(πόρτες, κάλυμμα, επένδυση)* στο πλαίσιο του οχήματος:
  - *Πλήρωμα 3a:* Τοποθέτηση πορτών και καλύμματος στο σκελετό του αυτοκινήτου.
  - *Πλήρωμα 3b:* Τοποθέτηση υπόλοιπης επένδυσης στο πλαίσιο του αυτοκινήτου.
- **Στάδιο 4:** τοποθέτηση τους τροχών:
  - *Πλήρωμα 4a:* Τοποθέτηση δύο μπροστινών ροδών.
  - *Πλήρωμα 4b:* Τοποθέτηση των δύο πίσω ροδών.
- **Στάδιο 5:** βάψιμο του SUV:
  - *Πλήρωμα 5a:* Βάψιμο από το πλάι του SUV.
  - *Πλήρωμα 5b:* Βάψιμο από πάνω του SUV.

Μείωση χρόνου σταδίου και αύξηση σταδίων (1/2)

- Σε προηγούμενο παράδειγμα είδαμε διασωλήνωση με **4 στάδια** με χρόνο σταδίου **1ns.**
- Αν μπορούμε να διαιρέσουμε τις λειτουργίες κάθε σταδίου στη μέση τότε μπορούμε να δημιουργήσουμε μια διασωλήνωση με **8 στάδια** και χρόνο σταδίου **0,5ns.**
- Αυτό θα αυξήσει το ρυθμό ολοκλήρωσης εντολών. *(π.χ. για 4 στάδια 1ns, υπάρχει ολοκλήρωση εντολής κάθε 1ns, με 8 στάδια 0,5 ns, υπάρχει ολοκλήρωση εντολής κάθε 0,5 ns).*
- **Δε μπορούμε να αυξάνουμε συνεχώς τον αριθμό** των σταδίων και ταυτόχρονα να μειώνουμε το χρόνο.

Μείωση χρόνου σταδίου και αύξηση σταδίων (2/2)

Σημαντικό στοιχείο της διασωλήνωσης:

Η διάρκεια κάθε σταδίου πρέπει να είναι ισορροπημένη με τις υπόλοιπες διάρκειες.

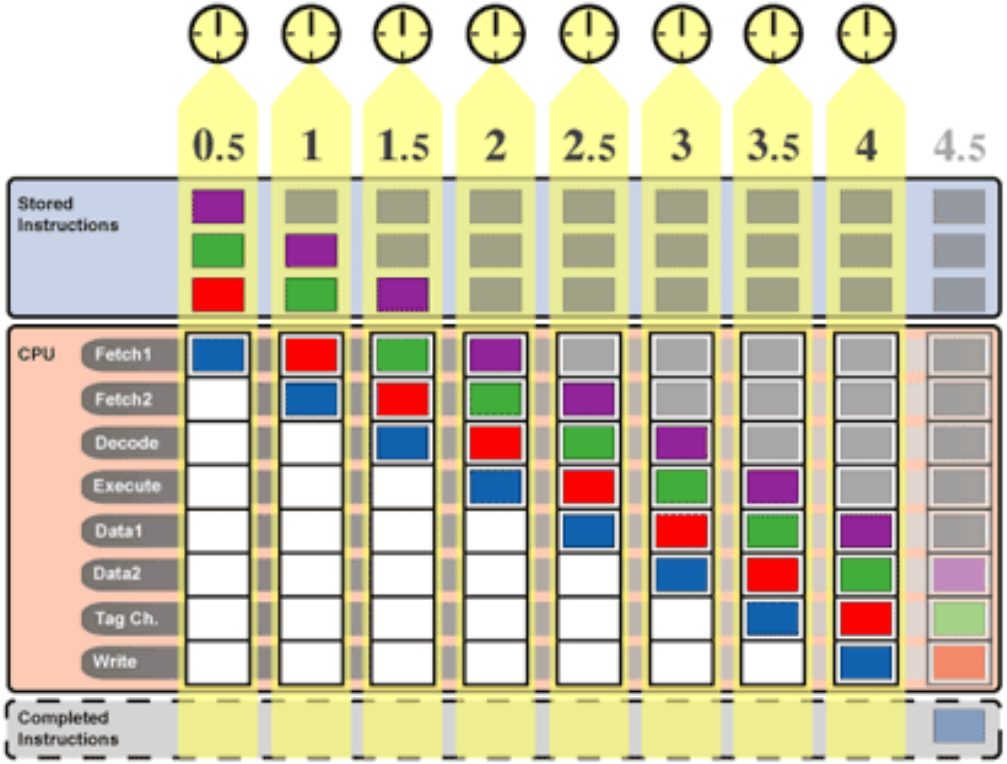
Διαφορετικά, υπάρχει χαμηλή απόδοση.

Αν η διάρκεια είναι ίσα, τότε ο χρόνος σταδίου υπολογίζεται από:

$$\frac{\chi\rho\acute{o}\nu\text{ος ανά εντολή}}{\text{αριθμό σταδίων διασωλήνωσης}}$$

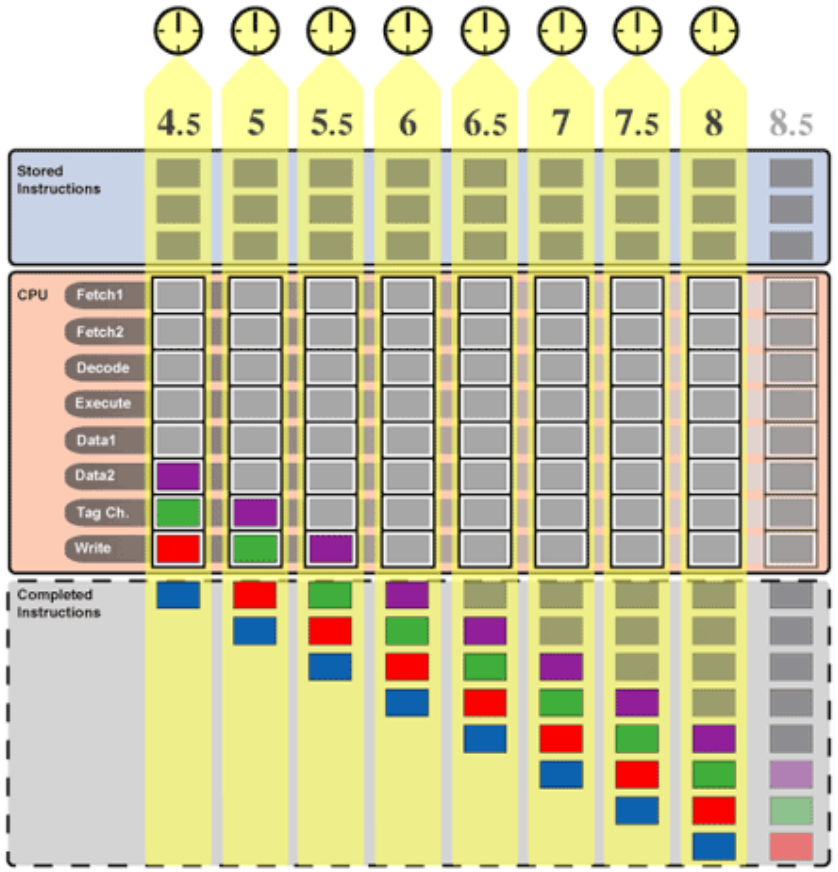
- Η επιτάχυνση στην απόδοση τότε είναι ίση με τον αριθμό των σταδίων.

Ένας υπολογιστής με διασωλήνωση σταδίου 0,5ns



Ολοκλήρωση εντολής κάθε 0,5 ns.

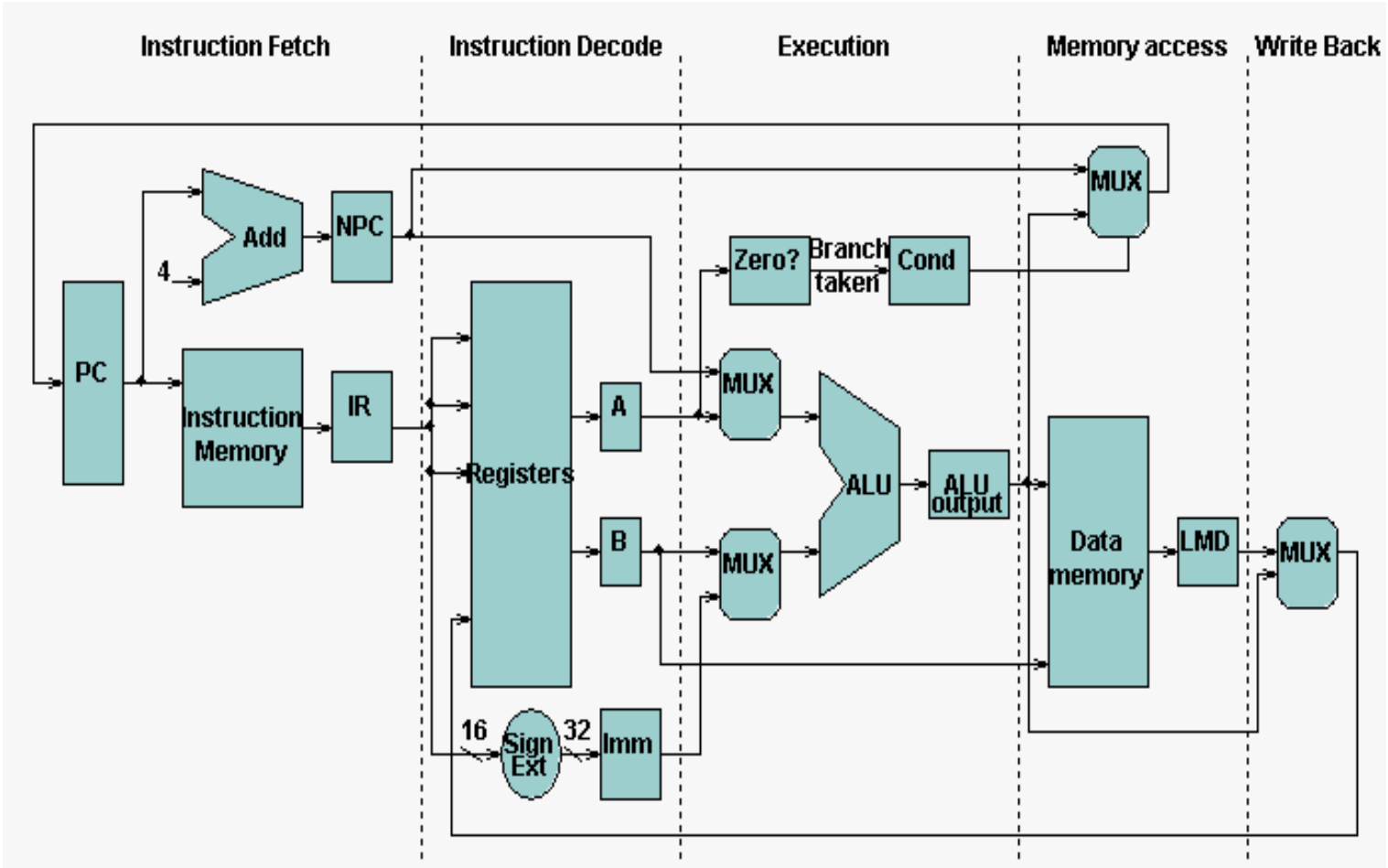
Ένας υπολογιστής με διασωλήνωση 0,5 ns



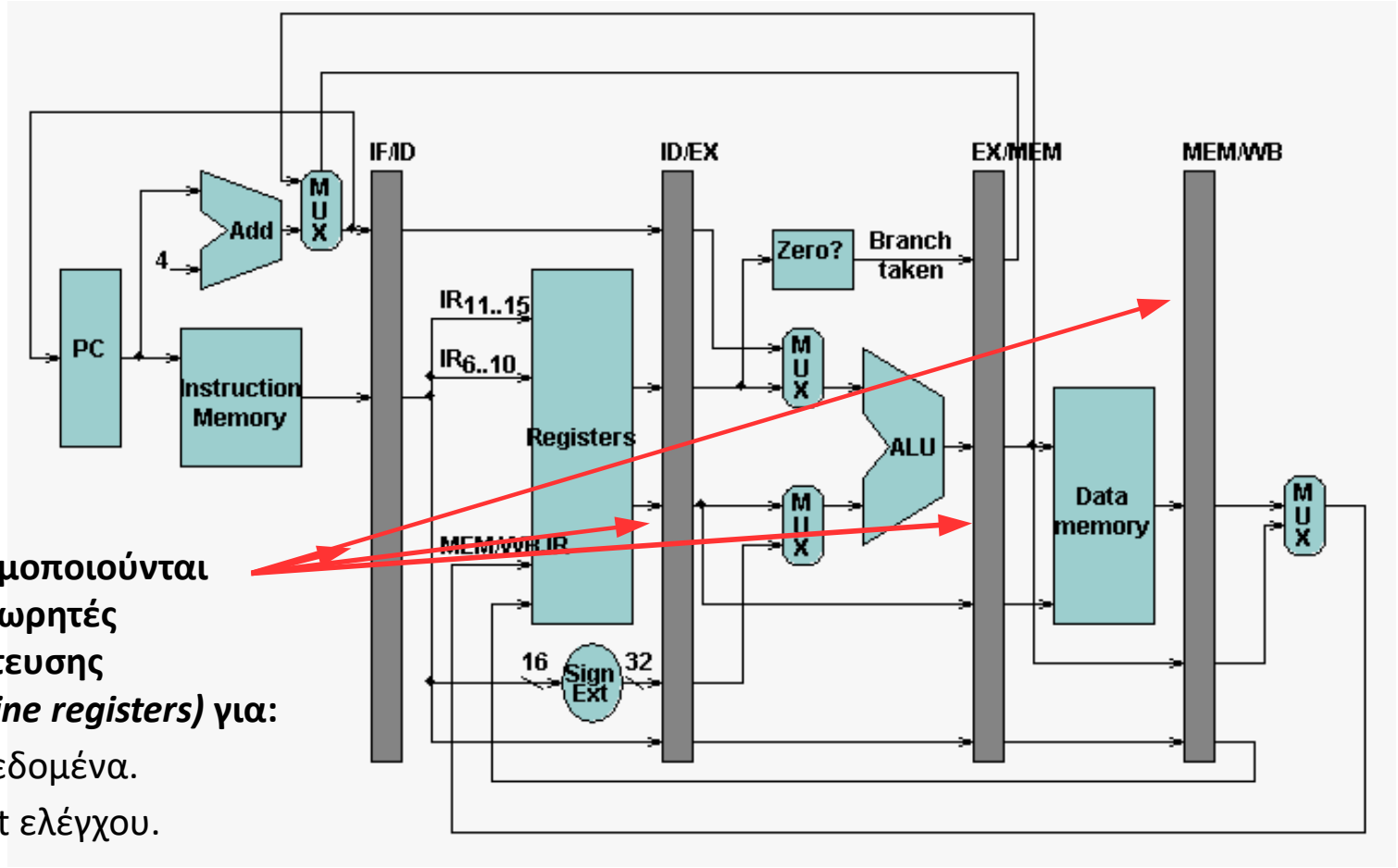
Έχει μειωθεί ο χρόνος κάθε σταδίου..

- Σε 4 στάδια με 1ns για 9ns είχαμε 5 εντολές.
- Σε 8 στάδια με 0,5ns για 9ns έχουμε 9 εντολές.

Δομικό Διάγραμμα datapath MIPS *(χωρίς διασωλήνωση)*



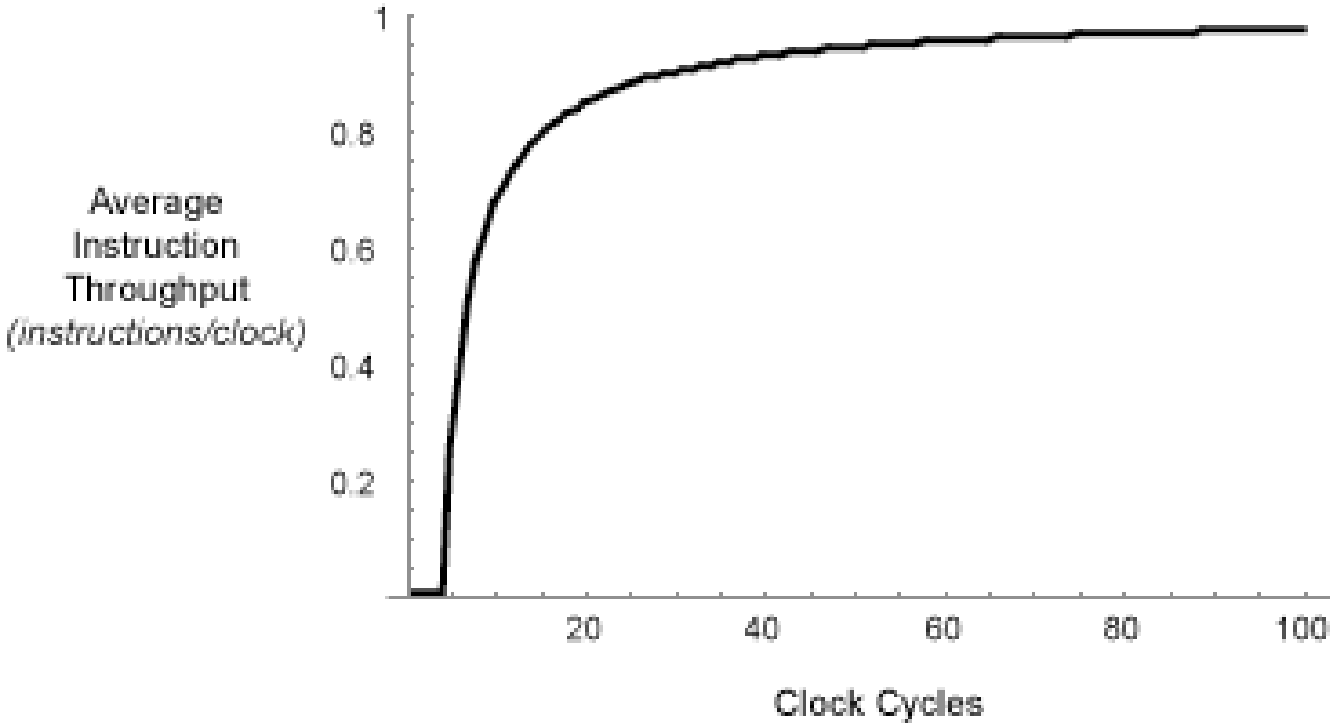
Δομικό Διάγραμμα datapath MIPS *(με διασωλήνωση 5 σταδίων)*



Χρησιμοποιούνται καταχωρητές διοχέτευσης *(pipeline registers)* για:

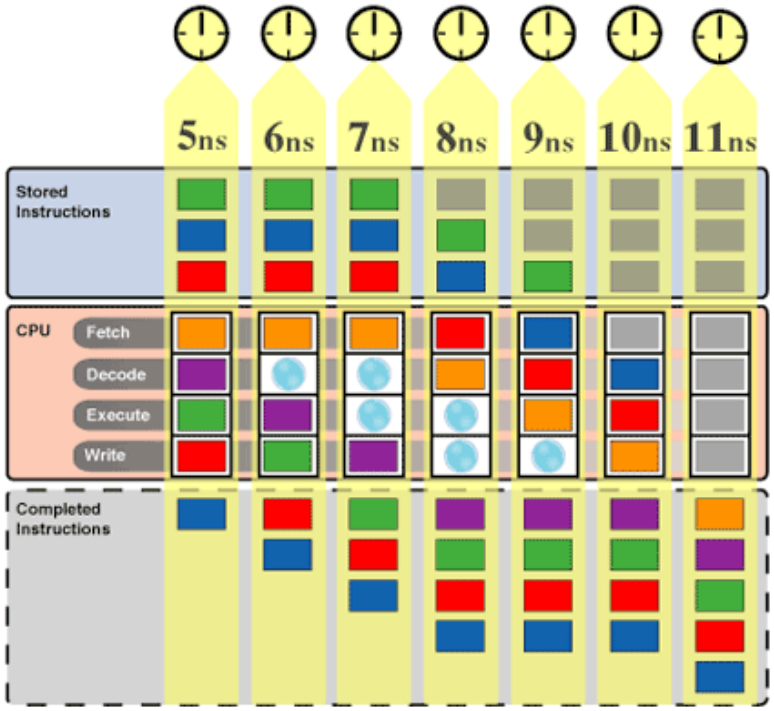
- Δεδομένα.
- Bit ελέγχου.

Η διασωλήνωση ως προς την απόδοση



Η διασωλήνωση δεν είναι πανάκεια

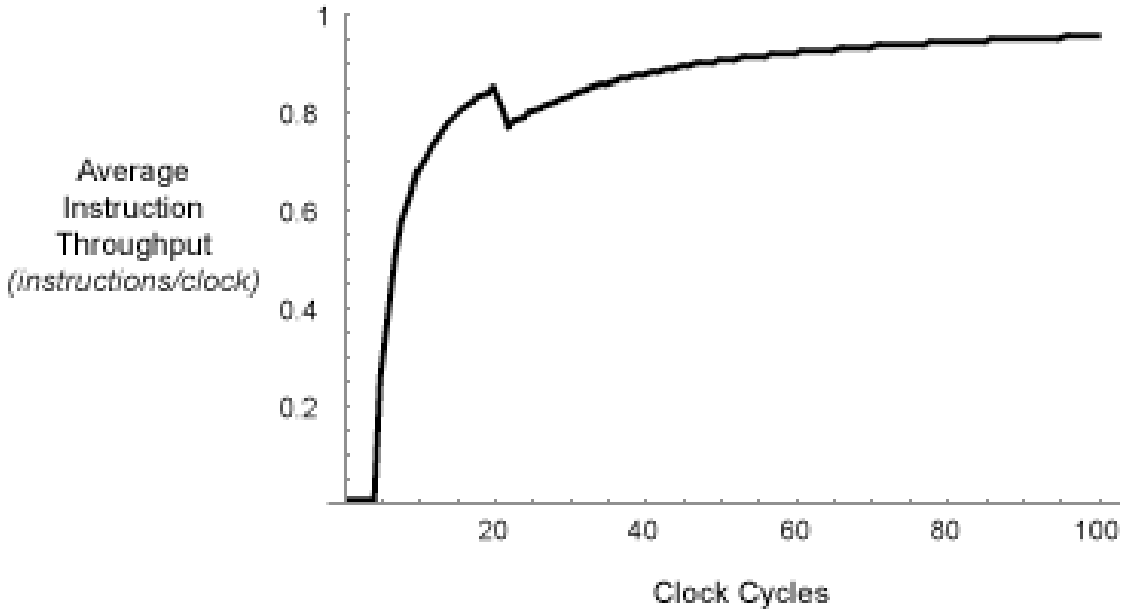
- Η διασωλήνωση προσθέτει αρκετή πολυπλοκότητα.
  ➔ Όλα θα πρέπει να είναι συγχρονισμένα.
- Μερικές φορές ένα στάδιο stalls *(“κολλάει”).* Όλη η σειρά σταματάει *(προσθέτονται φυσαλίδες)* μέχρι να συνεχίσει το στάδιο αυτό.
- Οι φυσαλίδες υλοποιούνται ως πράξη nop *(no operation).*



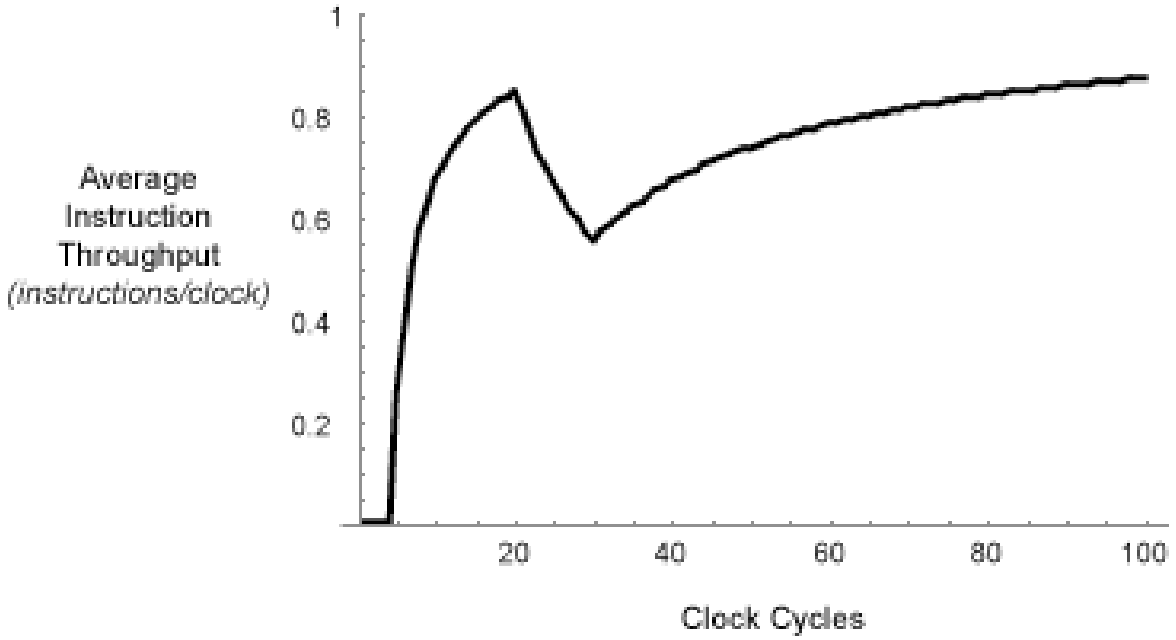
Ένα στάδιο μπορεί να “κολλήσει” δηλαδή, να καθυστερήσει η ολοκλήρωσή του, για ποικίλους λόγους, όπως αναλύεται παρακάτω!

2 κύκλοι stall και μείωση της απόδοσης

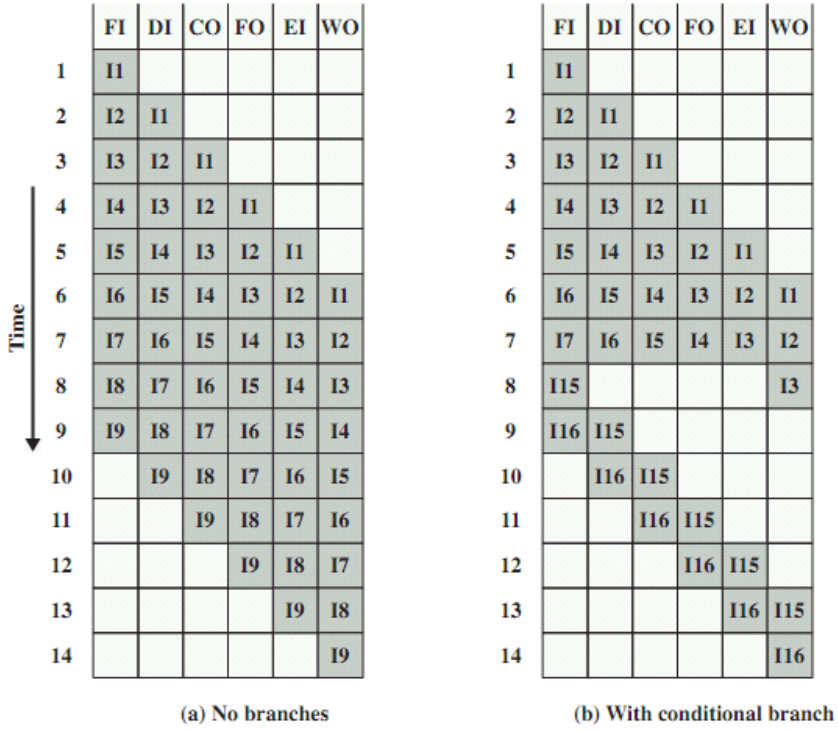
Κάθε φορά που δημιουργείται φυσαλίδα, τότε μειώνεται το CPI/IPC.



100 κύκλοι stall και σημαντική μείωση της απόδοσης



Μείωση της απόδοσης της διασωλήνωσης λόγω αλλαγής ροής



- Αν υπάρχει αλλαγή ροής εκτέλεσης, τότε αυτή θα γίνει γνωστή όταν ολοκληρωθεί η εκτέλεση της εντολής *(αναγράφεται ως WO – write operands).*
- Το αποτέλεσμα είναι να πρέπει να “αδειάσει” η διασωλήνωση και να “ξαναγεμίσει”.
- Στην εικόνα, η **I3** είναι εντολή αλλαγής ροής προς την εντολή **I15**.

Στάδια διασωλήνωσης και πολυπλοκότητα

- Δεν είναι ίδιας πολυπλοκότητας όλα τα στάδια της διασωλήνωσης.
- Το πιο αργό στάδιο της διασωλήνωσης καθορίζει και το συνολικό throughput.

Παράδειγμα:

- Το στάδιο fetch ολοκληρώνεται σε 0,7ns.
- Το στάδιο decode ολοκληρώνεται σε 0,5ns.
- Το στάδιο execute ολοκληρώνεται σε 1 ns.
- Το στάδιο write ολοκληρώνεται σε 0,8ns.

Ο χρόνος σταδίου καθορίζεται από το πιο αργό στάδιο, οπότε επιλέγεται ο χρόνος 1ns για όλα τα στάδια.

2 Σημαντικά στοιχεία της διασωλήνωσης

Καθυστερήσεις (Pipeline stalls):

Η καθυστέρηση σε ένα στάδιο διασωλήνωσης καθυστερεί όλη τη διασωλήνωση, και μειώνεται η απόδοση και ο ρυθμός ολοκλήρωσης εντολών του επεξεργαστή.

Γεμίσματα (Pipeline fills):

Το γέμισμα της διασωλήνωσης απαιτεί σημαντικό χρόνο, και μειώνει το ρυθμό ολοκλήρωσης εντολών και της απόδοσης. Όσο μεγαλύτερη είναι η διασωλήνωση, τόσο μεγαλύτερο πρόβλημα δημιουργείται.

Κίνδυνος διασωλήνωσης

- Κίνδυνος διασωλήνωσης εμφανίζεται όταν η διασωλήνωση ή ένα μέρος της, θα πρέπει να ακινητοποιηθεί, επειδή οι συνθήκες δεν επιτρέπουν τη συνέχιση της εκτέλεσης.

- Αυτή η ακινητοποίηση, αναφέρεται ως “φυσαλλίδα” *(pipeline bubble).*

Κίνδυνοι Διασωλήνωσης (1/2)

- Υπάρχουν καταστάσεις που ενδέχεται να προκαλέσουν προβλήματα στη διοχέτευση, γιατί θα οδηγήσουν σε λανθασμένους υπολογισμούς.
- Οι καταστάσεις αυτές δημιουργούνται από το γεγονός ότι πρέπει να ληφθεί μια απόφαση χωρίς να έχει ολοκληρωθεί η εκτέλεση μιας εντολής από την οποία εξαρτάται.

Για παράδειγμα:

```
cmp ax,bx
jne etiketa
```

- Δηλαδή, ξεκινάει η εκτέλεση της **`jne`** χωρίς να έχει ολοκληρωθεί η εκτέλεση της **`cmp`**.

Κίνδυνοι Διασωλήνωσης (2/2)

Κατηγορίες κινδύνων Διασωλήνωσης.

- **Κίνδυνοι ελέγχου** *(control hazards):*
  - πρέπει να ληφθεί μια απόφαση σχετικά με τη ροή εκτέλεσης του προγράμματος *(ποια είναι η επόμενη εντολή)* και δεν υπάρχουν διαθέσιμα τα στοιχεία.
  - η επόμενη εντολή δεν είναι διαθέσιμη *(π.χ. offchip ram).*
- **Κίνδυνοι δεδομένων** *(όταν υπάρχουν εξαρτήσεις):*
  - πρέπει να διαβαστεί ένας καταχωρητής στον οποίο δεν έχει γράψει η προηγούμενη εντολή τα αποτελέσματα *(read-after-write, RAW).*
  - Πρέπει να γραφεί ένας καταχωρητής, στον οποίο πρόκειται να διαβαστεί σε επόμενη εντολή *(write-after-read, WAR).*
  - Πρέπει να γραφεί ένας καταχωρητής, στον οποίο πρόκειται να γράψουμε σε επόμενη εντολή *(write-after-write, WAW).*
- **Δομικοί Κίνδυνοι** *(απαιτείται να χρησιμοποιηθεί το ίδιο στοιχείο/πόρος από δυο ή παραπάνω στάδια).*

Κίνδυνος ελέγχου ή διακλάδωσης

- Κάθε φορά που χρησιμοποιείται εντολή αλλαγής ροής εκτέλεσης, όπως:

```
add dl,bl
jmp again
```

- Κάθε φορά που καλείται μια συνάρτηση, όπως:

```
add dl,bl
call myfunctions
```

- Κάθε φορά που προκαλείται διακοπή, όπως:

```
mov ah,02
int 21h
```

Οι 3 μορφές κινδύνων δεδομένων

- **RAW ή πραγματική εξάρτηση:**
  Πρόβλημα γιατί δεν έχουν γραφεί τα αποτελέσματα όταν θα διαβαστούν.

```
add al,bl
add dl,al
```

Παράδειγμα: ο AL γράφεται και μετά διαβάζεται

- **WAR ή αντιεξάρτηση:**
  Πρόβλημα γιατί δε ξέρουμε αν η εντολή εγγραφής εκτελεστεί πιο γρήγορα.

```
add al,bl
mov bl,10
```

Παράδειγμα: ο bl διαβάζεται και μετά γράφεται

- **WAW ή εξάρτηση εξόδου:**
  Πρόβλημα γιατί δε ξέρουμε ποια εντολή εγγραφής θα εκτελεστεί πιο γρήγορα.

```
add al,bl
mov al,10
```

Παράδειγμα: ο al γράφεται και μετά πάλι γράφεται

Παραδείγματα κινδύνων δεδομένων

- Κίνδυνος WAW *(εγγραφή ύστερα από εγγραφή)*
- Η 2η εντολή ολοκληρώνεται πιο γρήγορα και γράφει τα αποτελέσματα στον R1. Η πρώτη εντολή, μετά από καθυστέρηση λόγω πρόσβασης στη μνήμη, γράφει τα αποτελέσματα στον R1. Είναι πρόβλημα, γιατί Ο R1 τελικά θα έχει τα αποτελέσματα της πρώτης εντολής και όχι της 2ης.

LW R1, 0(R2)	IF	ID	EX	MEM1	MEM2	WB
ADD R1, R2, R3		IF	ID	EX	WB	

- Κίνδυνος WAR *(εγγραφή ύστερα από ανάγνωση).*
- Η 2 εντολή τροποποιεί τον R2, ενώ η 1η εντολή τον χρησιμοποιεί για να δεικτοδοτούμενη πρόσβαση στην εξωτερική μνήμη *(και ως εκ τούτου με σημαντική καθυστέρηση)*. Είναι πρόβλημα γιατί ο R2 χρησιμοποιείται ακόμη.

SW R1, 0(R2)	IF	ID	EX	MEM1	MEM2	WB
ADD R2, R3, R4		IF	ID	EX	WB	

Παράδειγμα δομικών κινδύνων

- **Παράδειγμα1**: Το στάδιο fetch και το στάδιο write-back απαιτούν πρόσβαση στη μνήμη (αντιμετωπίζεται με χρήση κρυφών μνημών ή καθυστέρηση).
- **Παράδειγμα2**: το αρχείο καταχωρητών έχει μόνο μια πόρτα εγγραφής/ανάγνωσης, έτσι όταν μια εντολή προσπαθεί να γράψει τα αποτελέσματα (write-back) και μια άλλη να φέρει τις παραμέτρους (fetch).
- **Παράδειγμα3**: η εντολή **mov al,[si+bp]** απαιτεί να υπολογιστεί η δ/νση στο στάδιο εκτέλεσης και στη συνέχεια να γίνει μεταφορά με ενδεχόμενη καθυστέρηση στο ταυτόχρονο στάδιο fetch.

Αντιμετώπιση κινδύνων διασωλήνωσης (1/3)

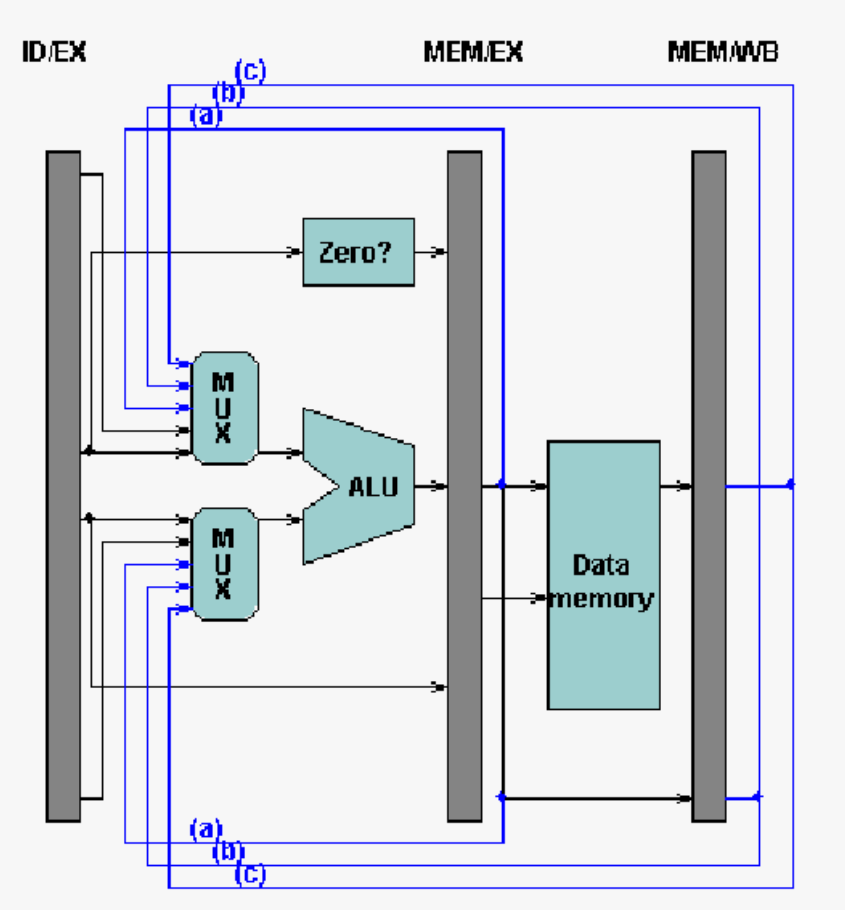
- Ο σχεδιαστής μπορεί για να μειώσει το κόστος να μην αντιμετωπίζει τους κινδύνους και απλά να καθυστερεί η διασωλήνωση (stall).
- Οι κίνδυνοι ελέγχου αντιμετωπίζονται με:
  - πρόγνωση διακλαδώσεων.
- Οι κίνδυνοι δεδομένων αντιμετωπίζονται με:
  - Διαίρεση της πρόσβασης σε καταχωρητή με εγγραφή στο πρώτο μισό του κύκλου, και ανάγνωση στο δεύτερο μισό.
  - Με μετονομασία καταχωρητών.
  - Με εγγραφή σε καταχωρητή μόνο στο τελευταίο στάδιο.
  - Με την τεχνική της προώθησης (forwarding), δηλαδή με την αντιγραφή των δεδομένων από ένα στάδιο σε προηγούμενο με παράκαμψη του σταδίου write-back με ειδικό κύκλωμα.

Αντιμετώπιση κινδύνων διασωλήνωσης (2/3)

- Οι δομικοί κίνδυνοι αντιμετωπίζονται:
  - Με παροχή επιπλέον υπολογιστικών πόρων υλικού (π.χ. επιπρόσθετη ALU, πολλαπλές θύρες εισόδου/εξόδου στο αρχείο καταχωρητών, κ.α.).

Η τεχνική της προώθησης (1/2)

- Τα αποτελέσματα από το στάδιο EX (execute) ή από το στάδιο WB (writeback) τροφοδοτούνται πίσω στο EX αν υπάρχει κίνδυνος RAW.
- Χρησιμοποιούνται κατάλληλοι πολυπλέκτες στην είσοδο της ALU.



Η τεχνική της προώθησης (2/2)

- Αν ένας επεξεργαστής δε χρησιμοποιεί αυτή την τεχνική τότε μπορεί να χρησιμοποιηθεί ένας compiler για να δημιουργήσει ορθό κώδικα με την προσθήκη εντολών **nop** *(no operation)* ανάμεσα στους κινδύνους RAW.
- Για αυτό το λόγο ένας compiler συνδέεται στενά με μια αρχιτεκτονική. Μια τεχνική *(στο παράδειγμά μας η αντιμετώπιση προβλημάτων διασωλήνωσης),* μπορεί να υλοποιηθεί είτε στο hardware είτε στο software.

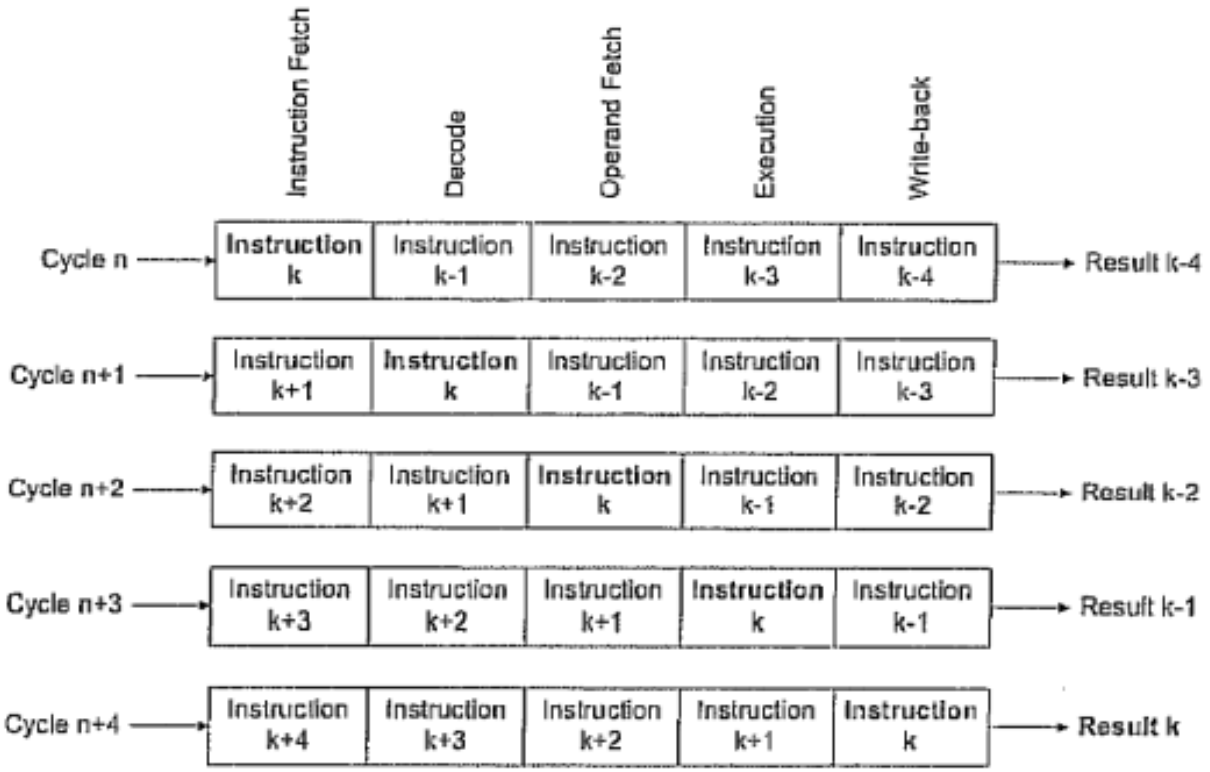
Αντιμετώπιση κινδύνων διασωλήνωσης (3/3)

- Δεν αντιμετωπίζονται όλοι οι κίνδυνοι αποτελεσματικά.
- Υπάρχουν περιπτώσεις που ο επεξεργαστής θα κάνει stall. Αυτό οφείλεται όταν:
  - χρησιμοποιείται ο ίδιος επεξεργαστής για αναγνώσεις/εγγραφές:

```
mov ax,[si]
sub si,5
xor si,si
...
```

  - Απαιτείται η λειτουργία μιας ενέργειας που δε μπορεί να ολοκληρωθεί στο χρόνο ενός σταδίου, όπως η πρόσβαση σε μνήμη ή I/O.
- Διακοπές, interrupts προκαλούν μείωση της απόδοσης.

Τα 5 στάδια διασωλήνωσης 386



Τα 5 στάδια διασωλήνωσης 80386/80486 (1/2)

- **Fetch:**
  - Οι εντολές από την κρυφή μνήμη ή από εξωτερική μνήμη εισέρχονται στον επεξεργαστή.
  - Τοποθετείται σε ένα από τα δύο ανεξάρτητα 32-byte προσωρινής μνήμης.
  - Γεμίζει την προσωρινή μνήμη με νέα δεδομένα αμέσως μόλις αδειάσει από τα παλιά δεδομένα.
  - Πολλαπλές εντολές έρχονται με κάθε ανάγνωση.
- **Αποκωδικοποίηση στάδιο 1:**
  - Λαμβάνει τις εντολές σε δύο παράλληλους αποκωδικοποιητές.
  - Καθορίζει αν οι δύο επόμενες εντολές είναι ζευγάρι.
  - Συγχρονίζει τις εντολές στις διασωληνώσεις U και V.
  - Παίρνει τις διευθύνσεις δεδομένων και παρέχει τα στοιχεία στο κύκλωμα πρόγνωσης διακλαδώσεων.

Τα 5 στάδια διασωλήνωσης 80386/80486 (2/2)

- **Αποκωδικοποίηση στάδιο 2:**
  - Καθορίζει τον τελεστή μνήμης σε ένα κύκλο ρολογιού.
  - Εκτελεί ελέγχους πρόσβασης στο τμήμα το οποίο βρίσκεται σε προστατευόμενη λειτουργία.
- **Εκτέλεση:**
  - ALU πράξεις, πρόσβαση στην κρυφή μνήμη, εγγραφή ενημέρωσης.
  - Έλεγχος πρόβλεψης διακλάδωσης για πληροφορίες και στις δύο διασωληνώσεις.
- **Writeback:**
  - Ενημέρωση καταχωρητών, σημαιών και κατάσταση επεξεργαστών.
  - Ελέγχονται ως προς την ορθότητα οι προβλέψεις διακλαδώσεων.

Οι κλασικές μηχανές RISC έχουν 5 στάδια *(clasic RISC pipeline)*

Τα βασικά στάδια διασωλήνωσης σε μια κλασσική αρχιτεκτονική RISC:

- **IF** = Instruction Fetch,
- **ID** = Instruction Decode,
- **EX** = Execute,
- **MEM** = Memory access,
- **WB** = Register write back.

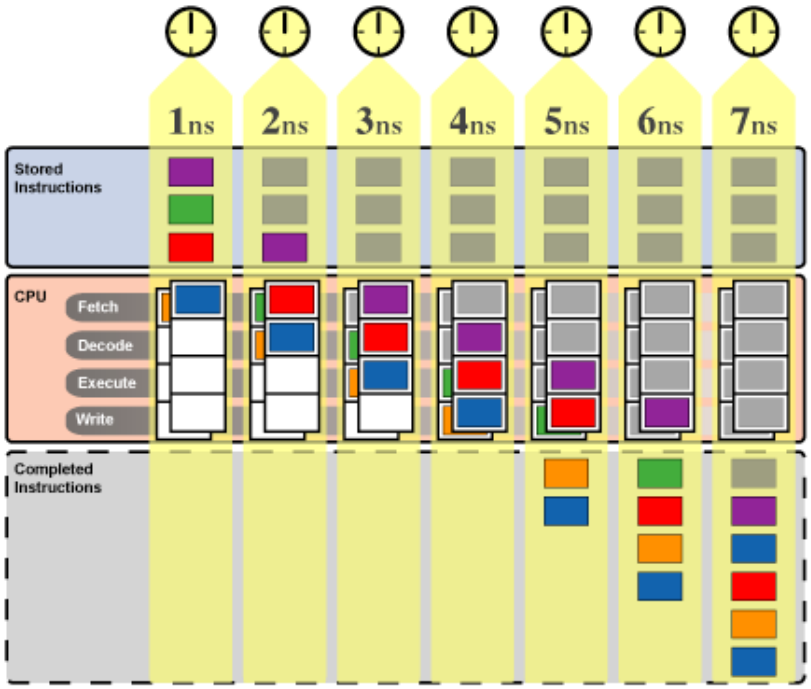
Στον κατακόρυφο άξονα, τοποθετούνται συνεχόμενες εντολές. Στον οριζόντιο άξονα βρίσκεται ο χρόνος. Σε μια δεδομένη στιγμή (π.χ. στην πράσινη στήλη) η πιο παλαιά εντολή βρίσκεται στο τελευταίο στάδιο (WB), ενώ η πιο νέα στο πρώτο στάδιο (IF).

Αριθμός σταδίων διασωλήνωσης και απόδοση

Δεν ισχύει όσα περισσότερα στάδια διασωλήνωσης, τόσο καλύτερη απόδοση:

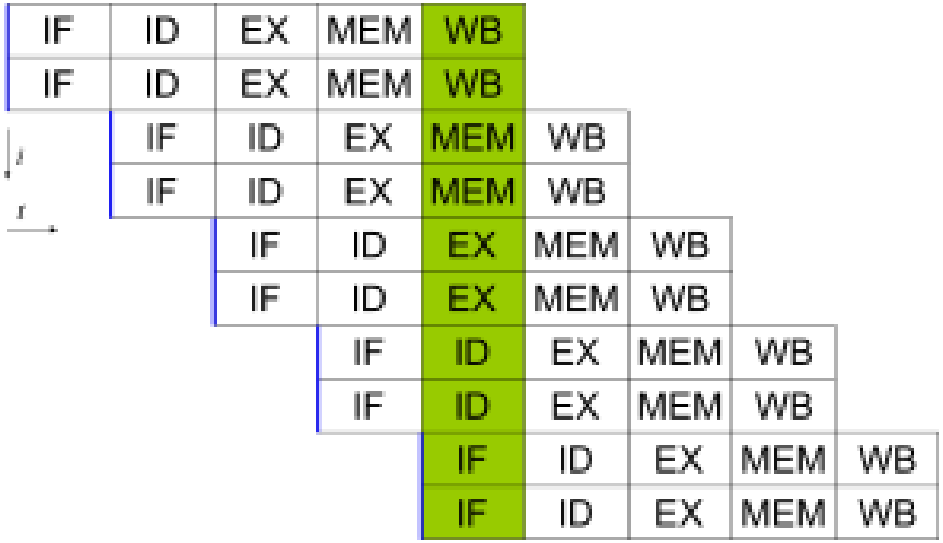
- Αν απαιτηθεί εκκαθάριση της διασωλήνωσης και έναυση από την αρχή το τίμημα είναι μεγάλο.
- Κάθε στάδιο συνδέεται με το επόμενο μέσω καταχωρητών. Προσθήκη πολλαπλών σταδίων, σημαίνει προσθήκη πολλαπλών καταχωρητών, οπότε αύξηση του χρόνου εκτέλεσης των εντολών.
- Όσα πιο πολλά στάδια διασωλήνωσης, τόσα πιο πολλά σήματα πρέπει να δημιουργηθούν, και άρα τόσο πιο πολύπλοκη και ενεργοβόρα η μονάδα ελέγχου.

Διασωλήνωση σε superscalar



- Υπάρχουν 2 διαφορετικές ουρές διασωλήνωσης.
- Υπάρχουν πολλαπλές ίδιες μονάδες στη διασωλήνωση.
- Απαιτείται run-time έλεγχος κατά το χρόνο εκτέλεσης *(run-time)* εξαρτήσεων δεδομένων.
- Υλοποιεί τον παραλληλισμό επιπέδου εντολής.

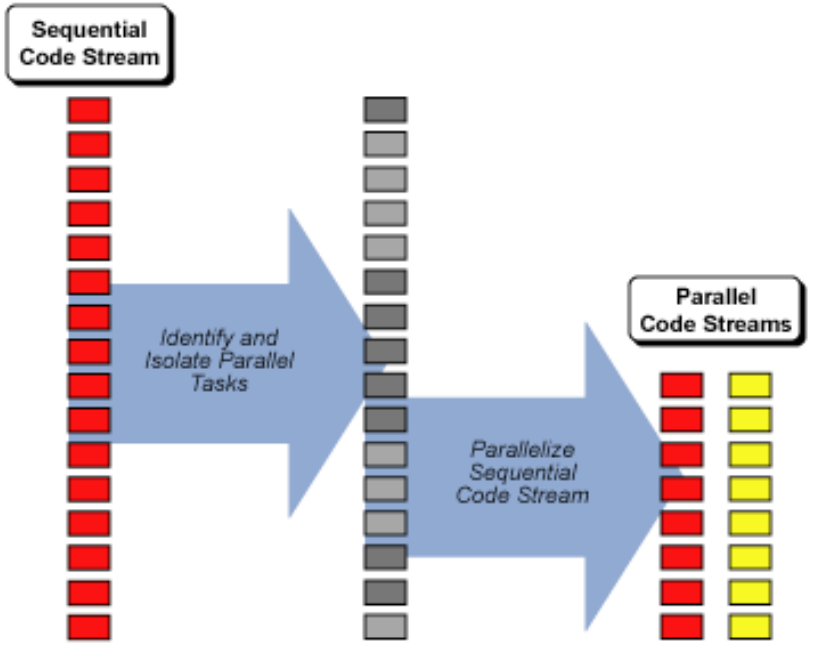
Ταυτόχρονο pipeline *(superscalar)*



- Διεκπεραιωτική ικανότητα περισσότερων εντολών σε κάθε στάδιο.
- Επιτυγχάνεται μέσω επιπρόσθετων μονάδων επεξεργασίας.

Simultaneous multithreading = hyperthreading *(SMT)*

- Υποστηρίζεται από τους σύγχρονους επεξεργαστές.
- Τα νήματα εκτελούνται παράλληλα.
- Υπάρχουν πολλαπλές ουρές διασωλήνωσης.
- Καλύτερη εκμετάλλευση του υλικού σε καθυστέρηση.



Hyperthreading *(SMT)*

- Λειτουργεί με τη δημιουργία **αντιγράφων αρχιτεκτονικής** τμημάτων του επεξεργαστή.
- Έτσι φαίνεται στο ΛΣ ότι υπάρχουν 2 εικονικοί επεξεργαστές για κάθε 1 φυσικό επεξεργαστή.
- Μόλις κάνει stall ένας εικονικός επεξεργαστής, συνεχίζει ο άλλος εικονικός *(στον ίδιο φυσικό επεξεργαστή).*
- Είναι διαφανής λειτουργία
  (γίνεται αυτόματα από τον επεξεργαστή).
- Το ΛΣ θα πρέπει να υποστηρίζει τους virtual processors ώστε να χρονοπρογραμματίζει κατάλληλα τα νήματα
  (π.χ. να μη χρονοπρογραμματίζει 2 νήματα στον ίδιο φυσικό επεξεργαστή).

Ερώτηση αυτοαξιολόγησης

Να εξηγηθεί γιατί η ταχύτητα μιας μηχανής που χρησιμοποιεί διασωλήνωση προσδιορίζεται από την καθυστέρηση που προκαλείται στη βραδύτερη βαθμίδα της!

Διευθυνσιοδότηση μνήμης και διασωλήνωση

- Οι σύγχρονοι επεξεργαστές αποφεύγουν ιδιαίτερες δ/σεις προκειμένου να μη καθυστερεί η διασωλήνωση.
- Υποστηρίζονται τα εξής χαρακτηριστικά:
  - Η προσπέλαση ενός ορίσματος δεν απαιτεί περισσότερες από μια προσπελάσεις μνήμης.
  - Μόνο οι εντολές φόρτωσης (load) και αποθήκευσης (store) προσπελαύνουν ορίσματα μνήμης.
  - Οι τρόποι διευθυνσιοδότησης, οι οποίοι χρησιμοποιούνται, δεν έχουν παρενέργειες.

Τέλος Ενότητας