

Processors (micro-processors)
Επεξεργαστές (μικρο-επεξεργαστές)

Επεξεργαστές γενικής χρήσης

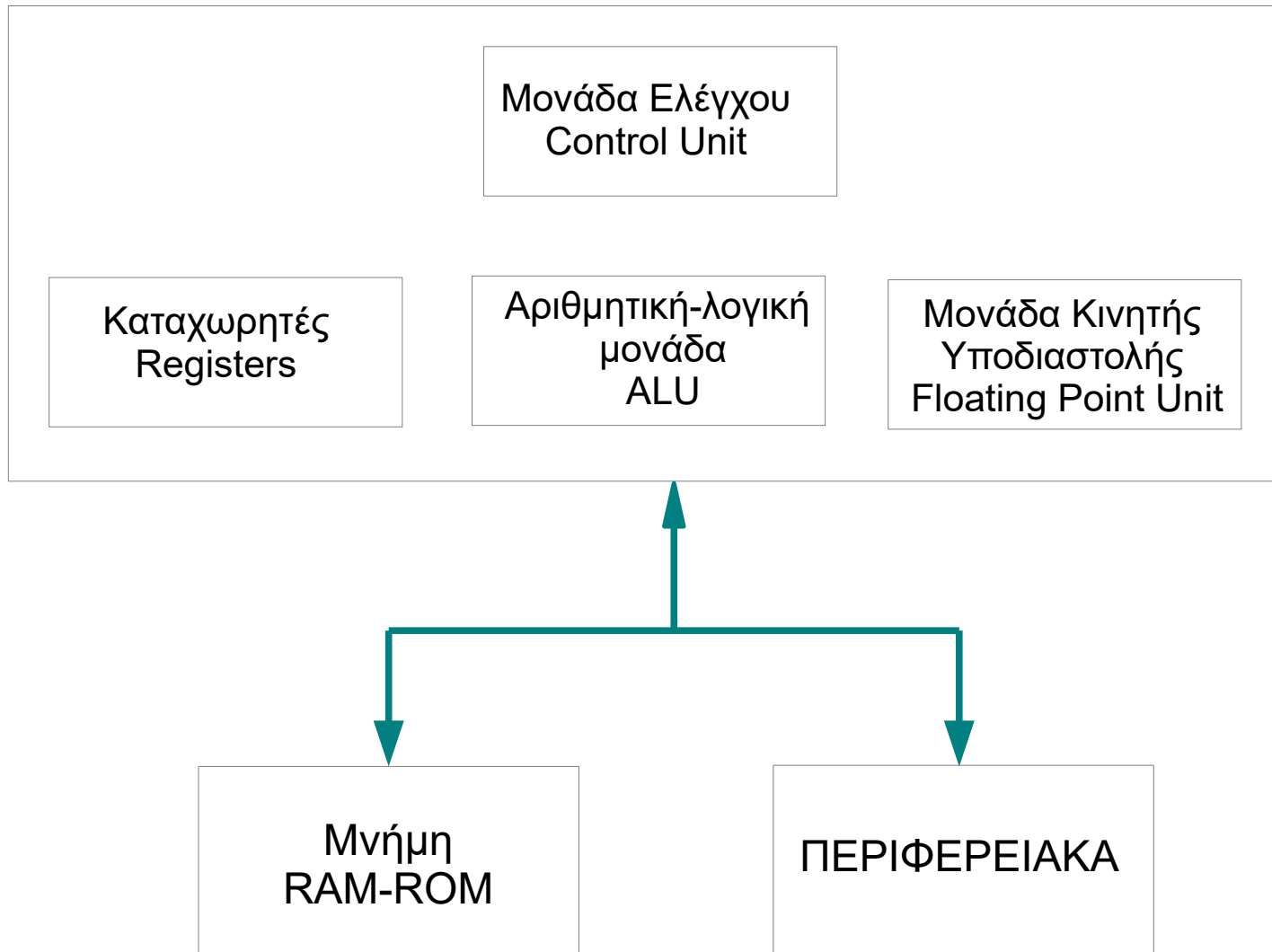
- Intel (x86, Pentium, Xeon, IA32, IA64)
- Apple–IBM–Motorola(AIM) PowerPC (PPC)

**Performance Optimization With Enhanced RISC –
Performance Computing**

2006 PowerISA

- Digital Alpha chip
- LSI SPARC (Sun)

- Motorola 6800
- MOS Technology 6502
- Intel 8080
- Zilog Z80



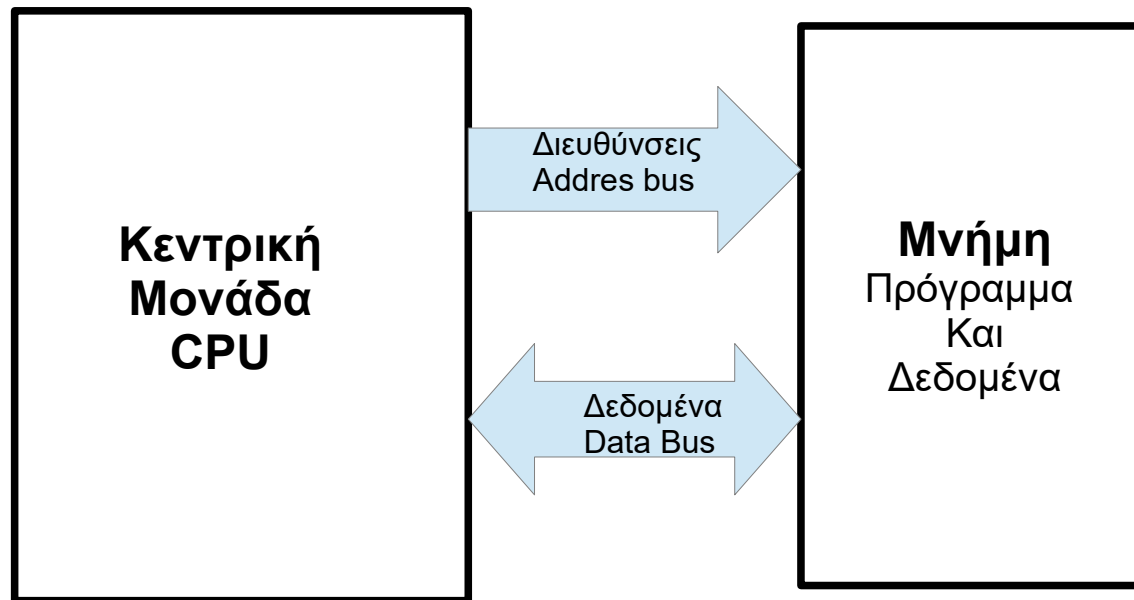
Ψηφιακοί επεξεργαστές σήματος- DSPs



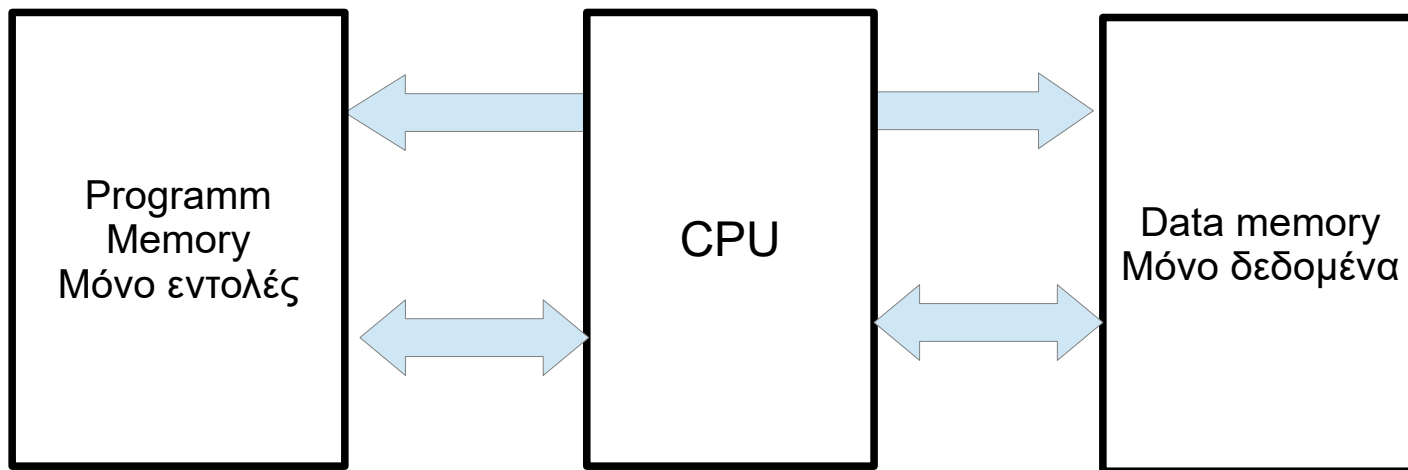
Επεξεργασία σε πραγματικό χρόνο

DSPs – Σχεδιασμένοι για αριθμητικούς υπολογισμούς

Αρχιτεκτονική Von Neumann

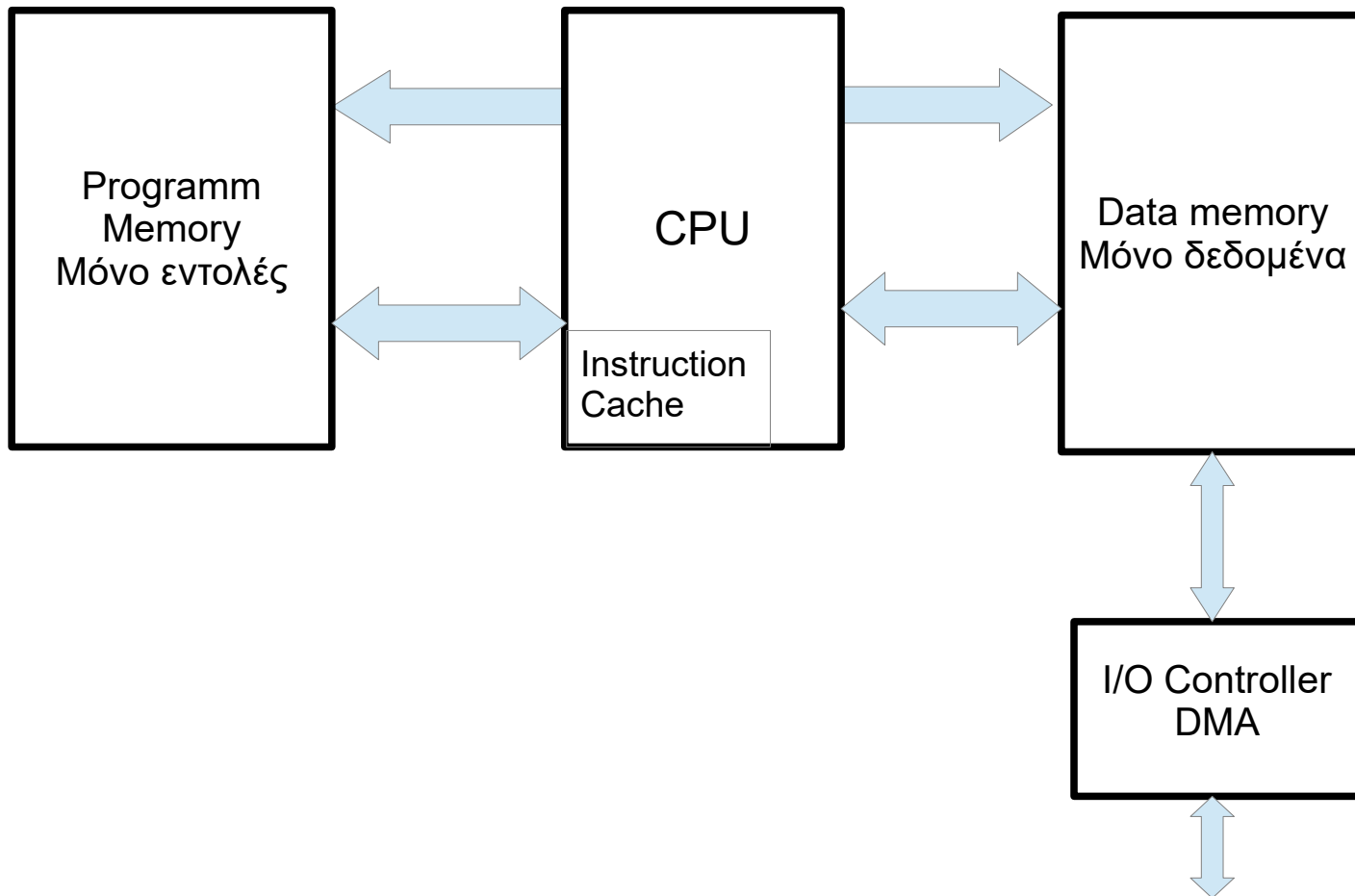


Αρχιτεκτονική Harvard



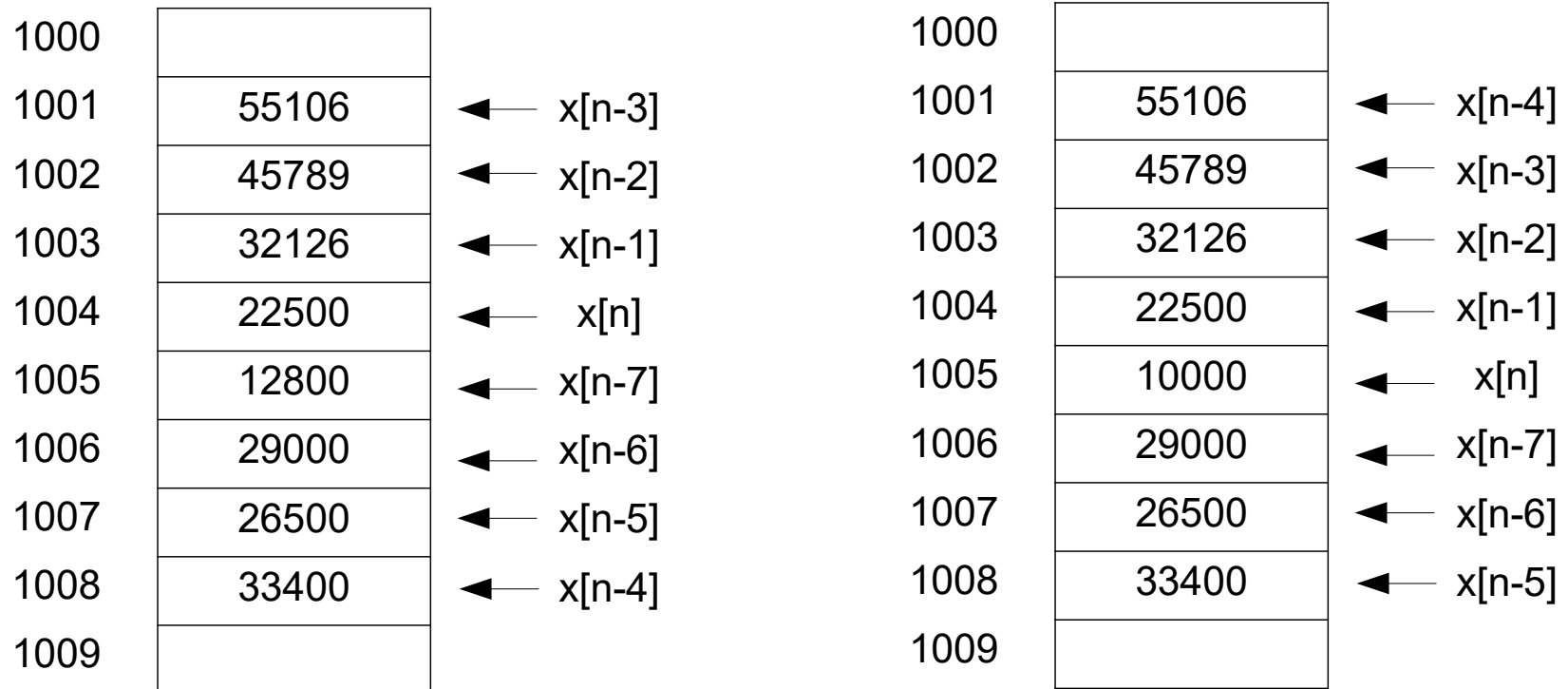
Ανεξάρτητοι δίαυλοι εντολών-δεδομένων
Μεγαλύτερη ταχύτητα από την Von Neumann

Super Harvard

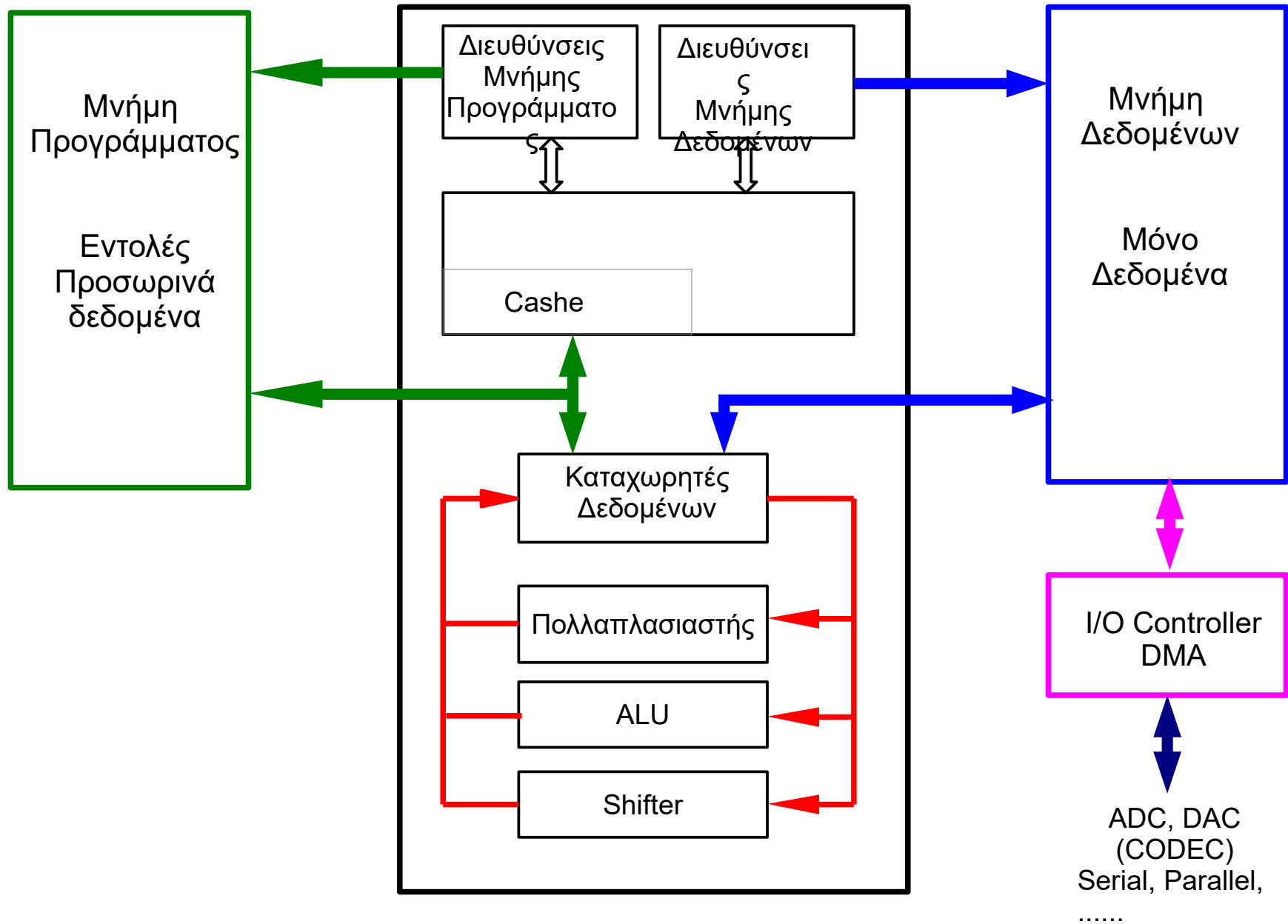


Κυκλικό buffer

Σε εφαρμογές πραγματικού χρόνου



Τυπικός DSP



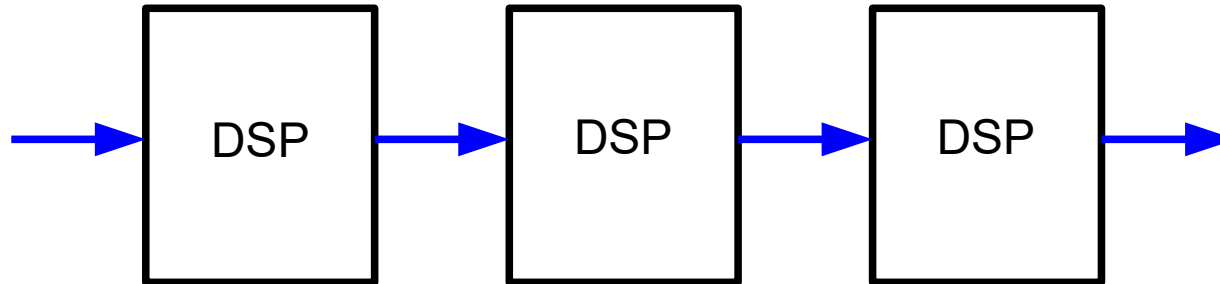
Επεξεργασία Ήχου- Εικόνας

- **Ηχος (High Fidelity)**
 - Εύρος ζώνης 20Hz – 20KHz
 - Ρυθμός δειγματοληψίας 44.1KHz
 - 2 κανάλια
 - 88200 δείγματα/sec
- **Τηλεόραση υψηλής ευκρίνειας (HDTV)**
 - 1920 x 1080 pixels
 - 3 χρώματα/pixel
 - 30 καρέ/sec
 - 186624000 δείγματα/sec
- **Τηλεόραση 4K**
 - 3840 x 2106 pixels
 - 3 χρώματα/pixel
 - 30 καρέ/sec
 - 746496000 δείγματα/sec

Πολυεπεξεργασία (multiprocessing)

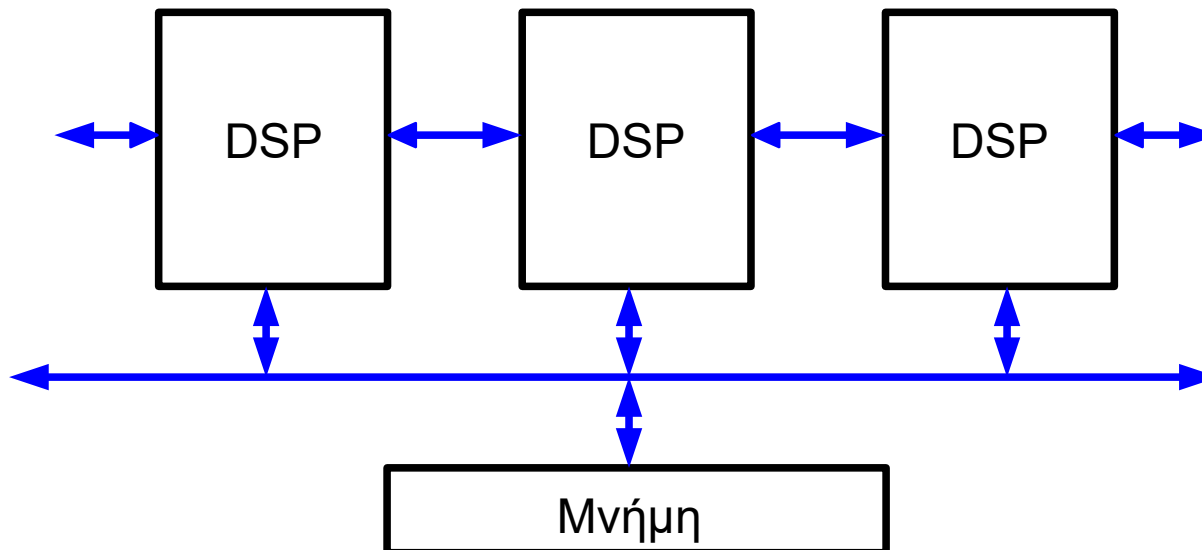
Ροή δεδομένων (Data Flow)

Ο αλγόριθμος χωρίζεται σε διαδοχικά βήματα

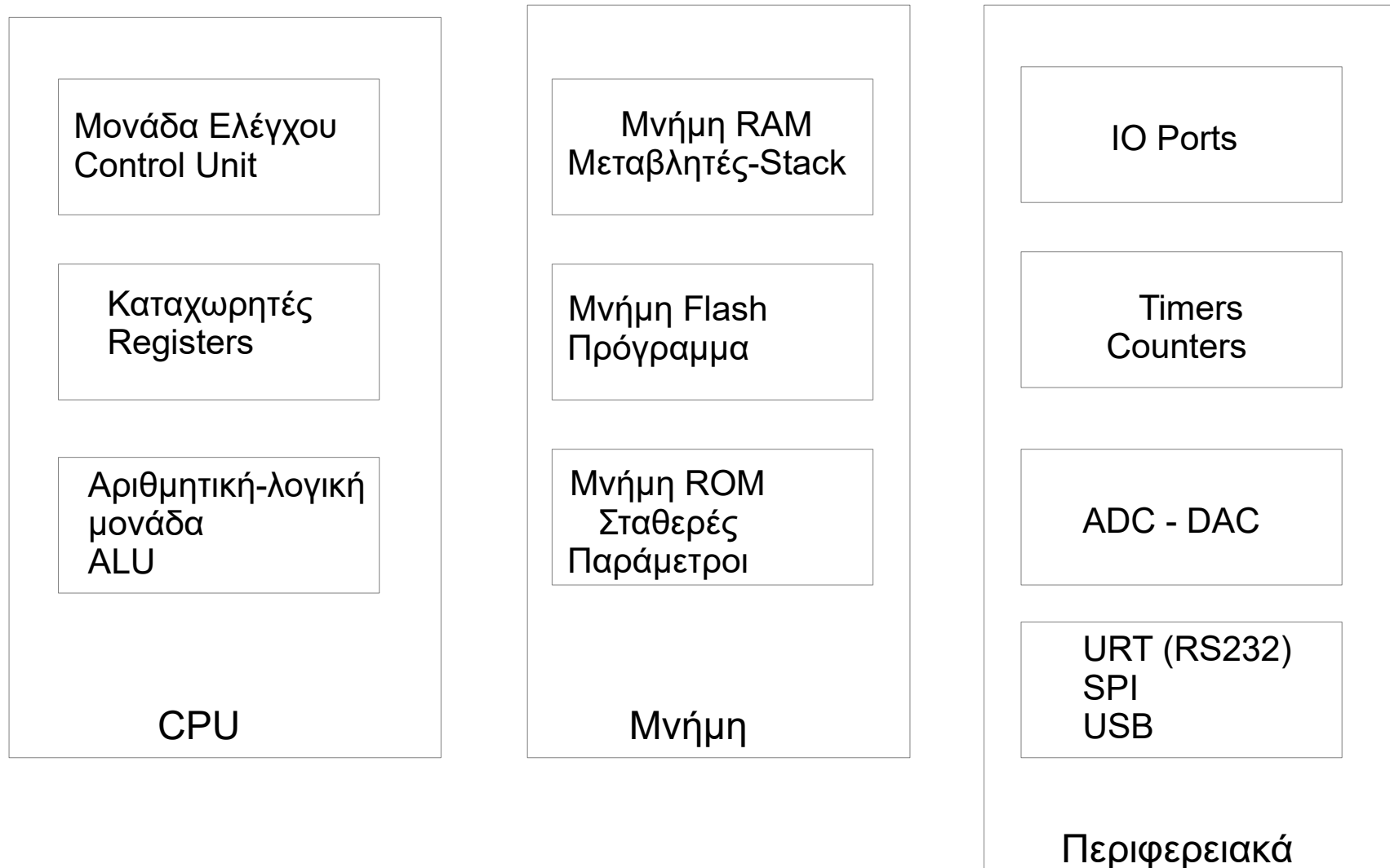


Σύμπλεγμα (Σμήνος) (Cluster)

Οι επεξεργαστές επικοινωνούν μέσω της κοινής μνήμης

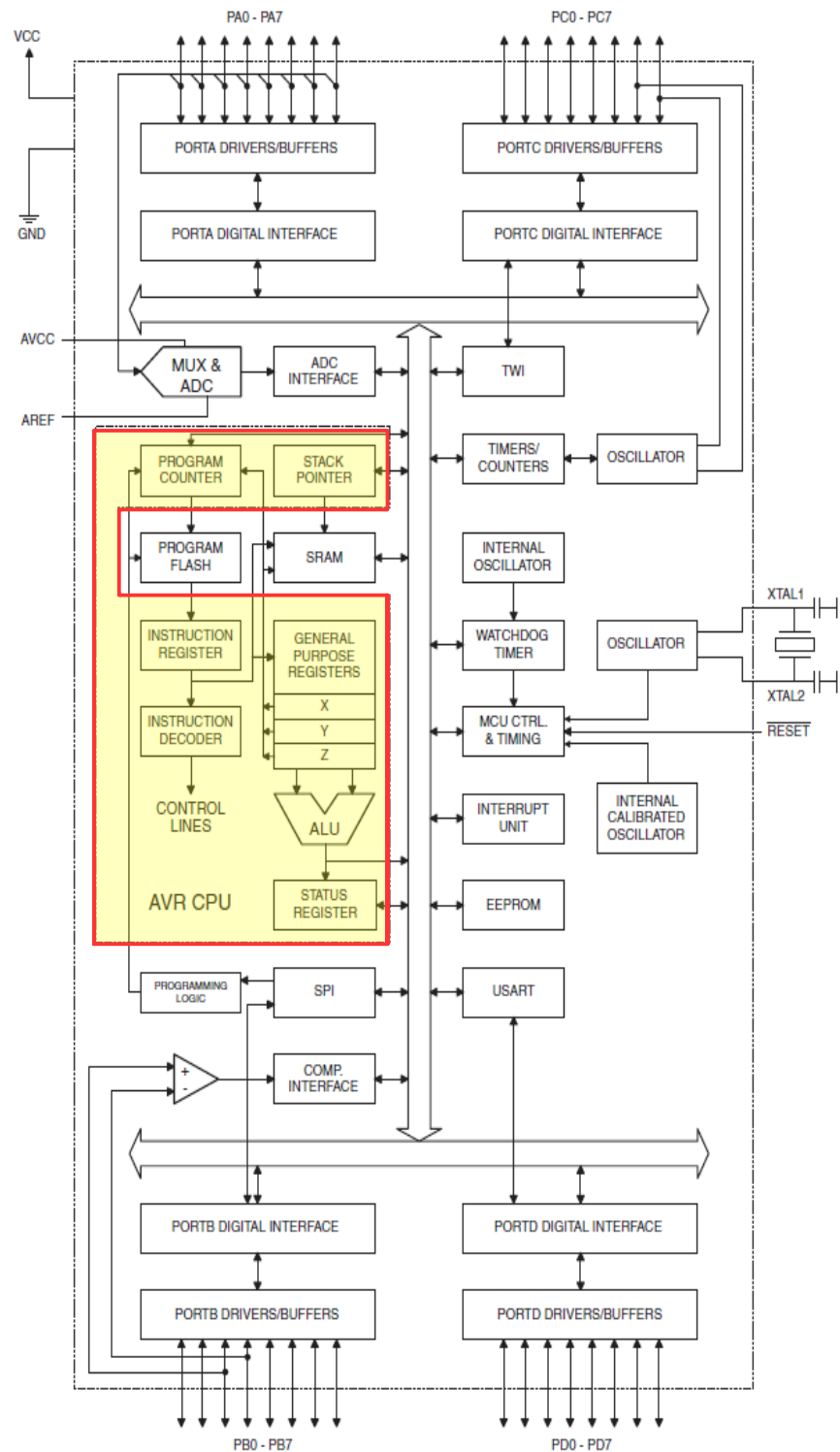


Μικροελεγκτές Microcontrollers μ Cs



ATMEL ATMega32

32KB Flash
2KB RAM
1KB ROM



Τύποι επεξεργαστών και εφαρμογές

Κύριες εφαρμογές κάθε τύπου επεξεργαστή

Επεξεργαστές γενικής χρήσης

Εφαρμογές :

- Υπολογιστικά συστήματα

Χαρακτηριστικά

- Μεγάλη μνήμη (address bus)
- Μεγάλο μήκος λέξης (32 , 64 bits)
- Αριθμητική μονάδα κινητής υποδιαστολής (FPU)
- Σύνθετοι τρόποι διευθυνσιοδότησης (addressing modes) – Υποστήριξη ανωτέρων γλωσσών προγραμματισμού
- Υψηλή κατανάλωση
- Μεγάλο κόστος

DSPs

Εφαρμογές

- Επεξεργασία σήματος σε πραγματικό χρόνο
- Τηλεπικοινωνίες
- Εγγραφή – Αναπαραγωγή ήχου και εικόνας
- Κάρτες γραφικών
- Συστήματα ελέγχου

Χαρακτηριστικά

- Αριθμητική σταθερής ή κινητής υποδιαστολής
- Αρχιτεκτονική προσαρμοσμένη για υψηλή ταχύτητα αριθμητικών υπολογισμών
- Μικρή μνήμη
- Μήκος λέξης 16 ή 32 bits
- Περιφερειακά (ADC, Ports, ...)
- Χαμηλή κατανάλωση

Μικροελεγκτές

Εφαρμογές

- Περιφερειακά υπολογιστών (Πληκτρολόγιο, ποντίκι, εκτυπωτές κλπ)
- Αυτοκίνητο (Controller Area Network , CAN Bus)
- Ηλεκτρικές συσκευές
- Τηλεχειριστήρια

Χαρακτηριστικά

- Μνήμη και περιφερειακά στο ίδιο ολοκληρωμένο
- Μικρή μνήμη
- Μικρός αριθμός τρόπων διευθυνσιοδότησης
- Λέξη 8 ή 16 bits
- Χαμηλή κατανάλωση
- Χαμηλό κόστος
- Οχι FPU

Προγραμματισμός

- **Επεξεργαστές γενικής χρήσης**
 - Κυρίως ανώτερες γλώσσες C,C++ (Λειτουργικό Σύστημα, Εφαρμογές)
 - Assembly (device drivers)
- **DSP**
 - Ανώτερες γλώσσες C,C++,Matlab
 - Assembly για βελτιστοποίηση του κώδικα
- **Embedded systems (microcontrollers)**
 - Assembly για κρίσιμες λειτουργίες ελέγχου
 - Ανώτερες γλώσσες C,Java

Στάδια εκτέλεσης εντολής

- Μεταφορά εντολής από τη μνήμη (Fetch)
- Αποκωδικοποίηση εντολής (Decode)
 - Ανάλυση εντολής σε μικροπρόγραμμα
- Εκτέλεση εντολής (Execute)
- Μεταφορά αποτελεσμάτων στους καταχωρητές της CPU (Write Back)

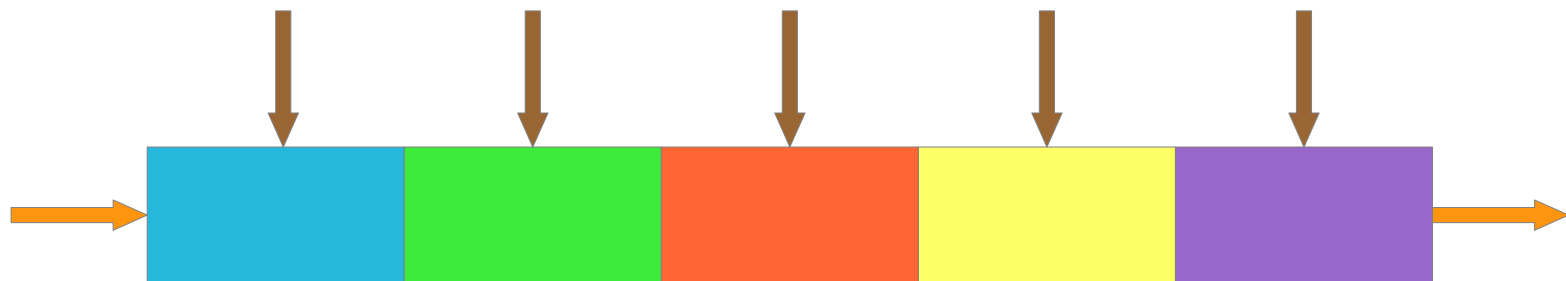
Κατά την εκτέλεση μιας εντολής οι επόμενες εντολές μεταφέρονται από τη μνήμη και αποκωδικοποιούνται (instruction pipeline). Οι σύγχρονοι επεξεργαστές έχουν pipelines πολλών σταδίων

Λογοι για τη πρόβλεψη διακλαδώσεων

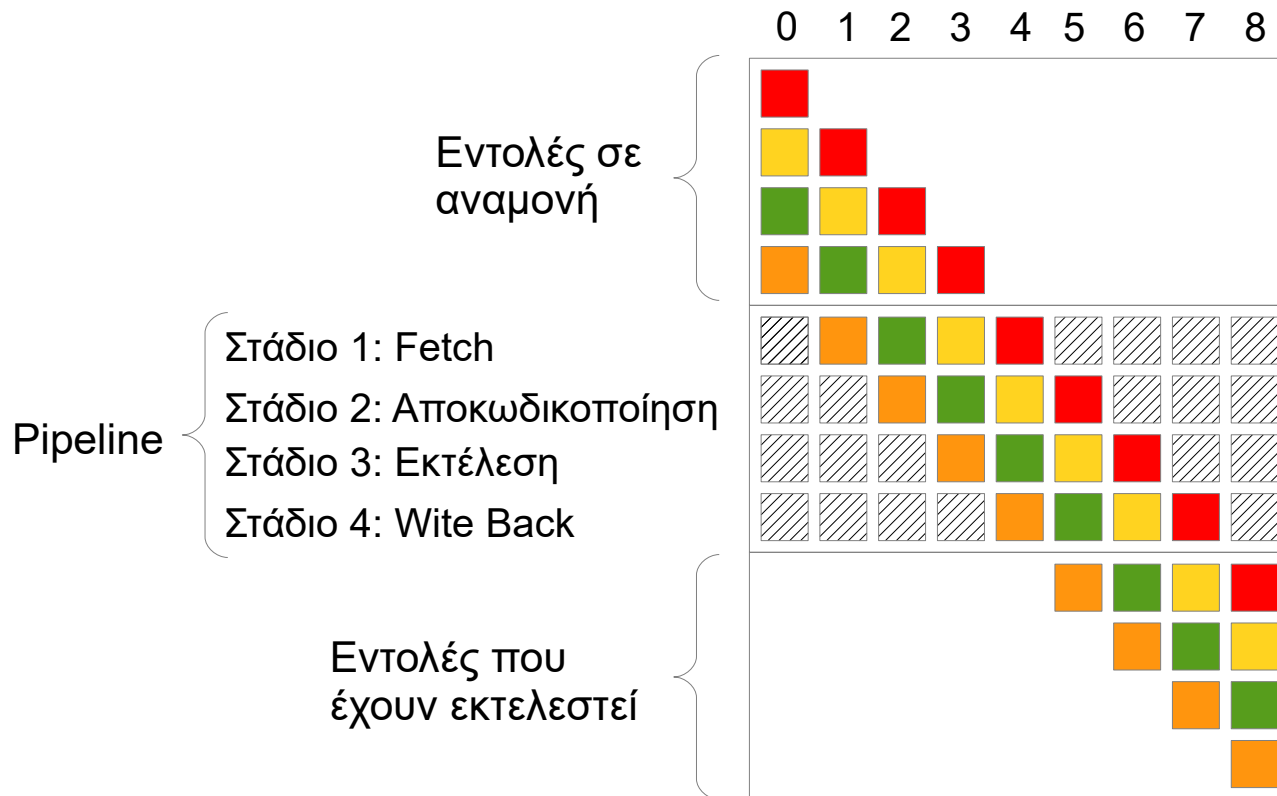
Pipeline

Ένα σύνολο διαδοχικών διαδικασιών. Η έξοδος του ενός σταδίου είναι η είσοδος του επόμενου.

Γραμμή Παραγωγής



Pipeline



Εντολές Διακλάδωσης (Branch)

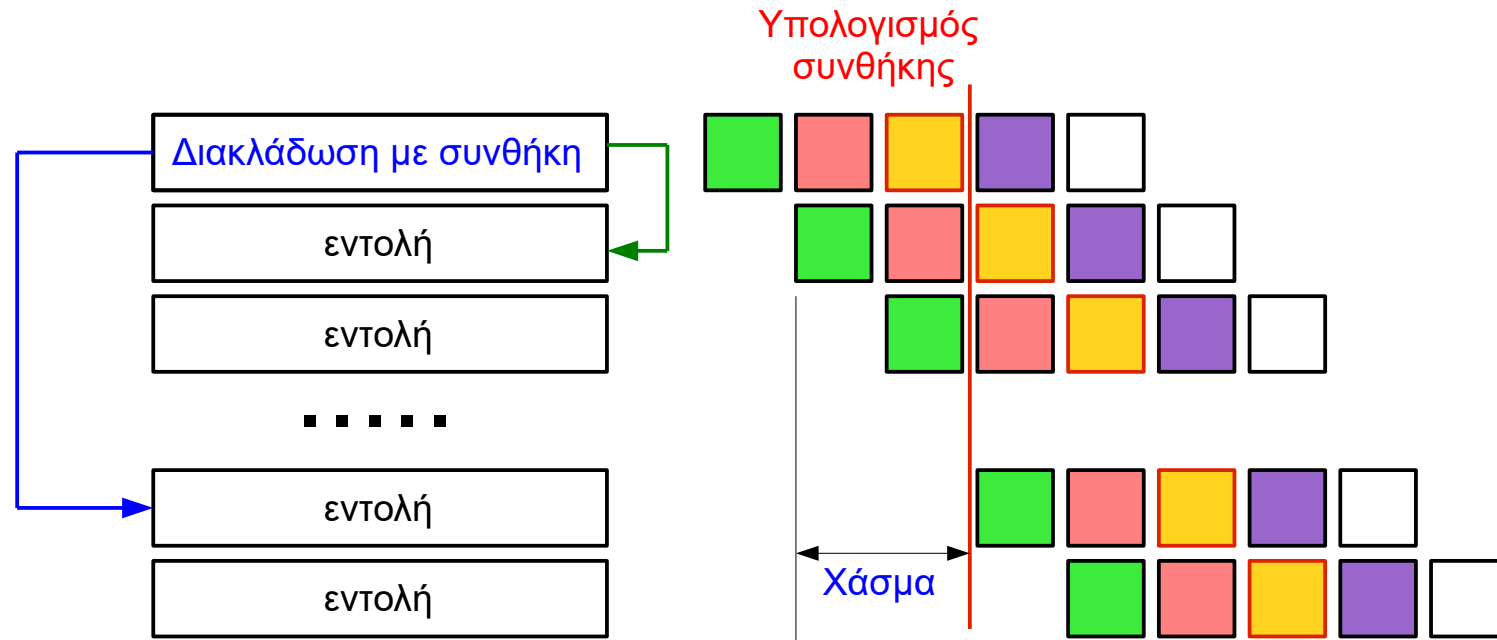
Τύποι εντολών διακλάδωσης

	Υπο συνθήκη	Χωρίς συνθήκη
Άμεσες	If-then-else for, while bez , bnz	Κλήση διαδικασίας (procedure, function call) goto
Έμμεσες		return function pointers

Πρόβλεψη διακλάδωσης Branch Prediction

Branch Predictor : Ένα κύκλωμα που προσπαθεί να προβλέψει το κλάδο που θα ακολουθήσει ένα πρόγραμμα μετά από μια εντολή διακλάδωσης.

1. Χάσμα στην pipeline



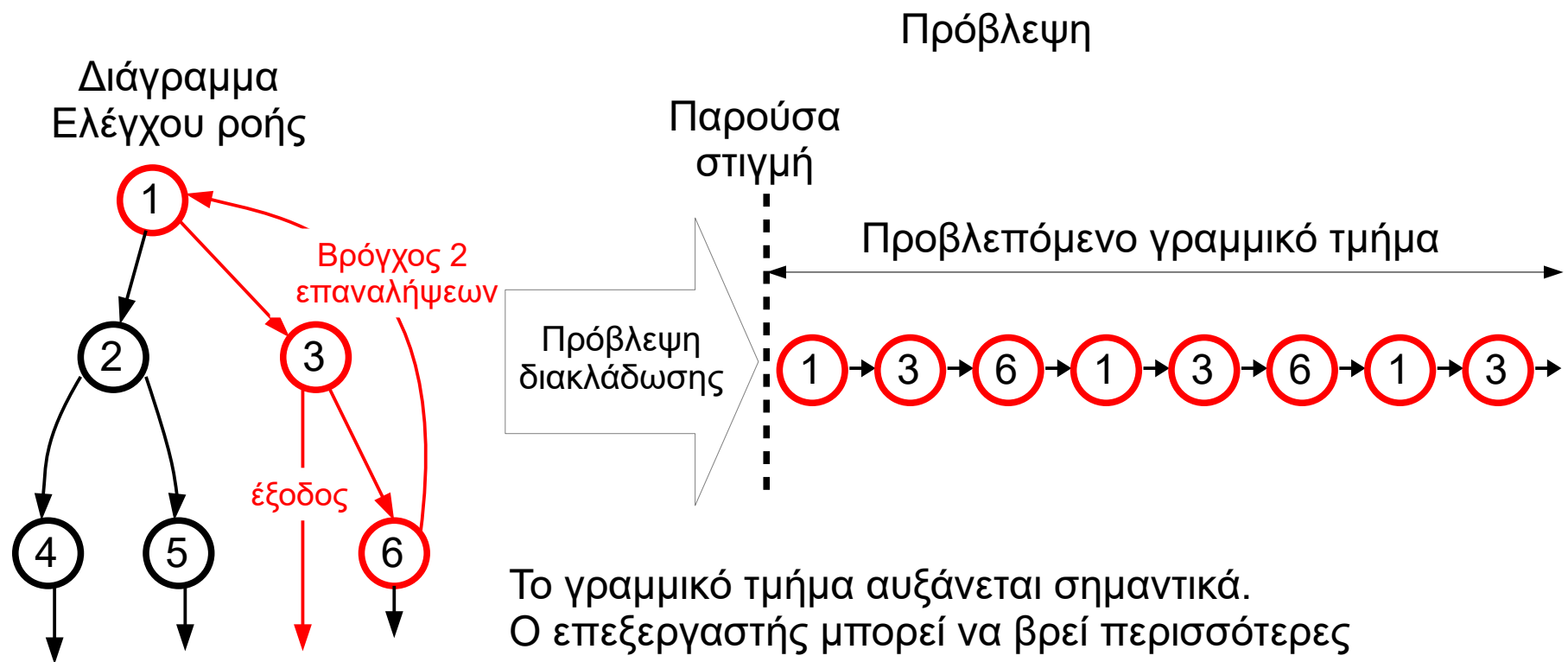
Άλμα έγινε (Branch Taken)

Άλμα δεν έγινε (Branch Not Taken)

Το χάσμα (ο χαμένος χρόνος) χωρίς ή λανθασμένη πρόβλεψη εξαρτάται από τον αριθμό των σταδίων της pipeline από το fetch stage to the execute stage

2. Παράλληλη εκτέλεση

- Ο επεξεργαστής μπορεί να διαπιστώσει αν οι εντολές μπορούν να εκτελεστούν παράλληλα αλλά το γραμμικό τμήμα (οι εντολές μεταξύ δύο διακλαδώσεων) είναι μικρό για να παρεμβληθούν άλλες εντολές (κάθε 5η – 8η εντολή είναι εντολή διακλάδωσης)
- Η πρόβλεψη διακλάδωσης μπορεί να αυξήσει το γραμμικό τμήμα.



Το γραμμικό τμήμα αυξάνεται σημαντικά.
Ο επεξεργαστής μπορεί να βρει περισσότερες εντολές που είναι δυνατόν να εκτελεστούν παράλληλα.

Γιατί είναι χρήσιμη η πρόβλεψη διακλάδωσης

- Οι εντολές διακλάδωσης εμφανίζονται συχνά στα προγράμματα.
 - 15% - 25% των εντολών είναι διακλαδώσεις
- Οι σύγχρονοι επεξεργαστές έχουν pipelines πολλών σταδίων
- Χωρίς πρόγνωση πολλοί κύκλοι λειτουργίας του επεξεργαστή δε αξιοποιούνται.
-

Πρόβλεψη διακλάδωσης

- Πρόβλεψη του αποτελέσματος μιας εντολής διακλάδωσης
 - Θα γίνει (Taken)
 - Δεν θα γίνει (Not Taken)
- Διεύθυνση προορισμού
 - Καταχωρητής διευθύνσεων + μήκος εντολής : Δεν θα γίνει
 - Καταχωρητής διευθύνσεων + απόσταση : Θα γίνει
- Πρόβλεψη διεύθυνσης προορισμού
 - Branch Target Buffer (BTB)

Στατική πρόβλεψη (software)

Περιπτώσεις που η διεύθυνση διακλάδωσης μπορεί να προβλεφθεί με μεγάλη πιθανότητα.

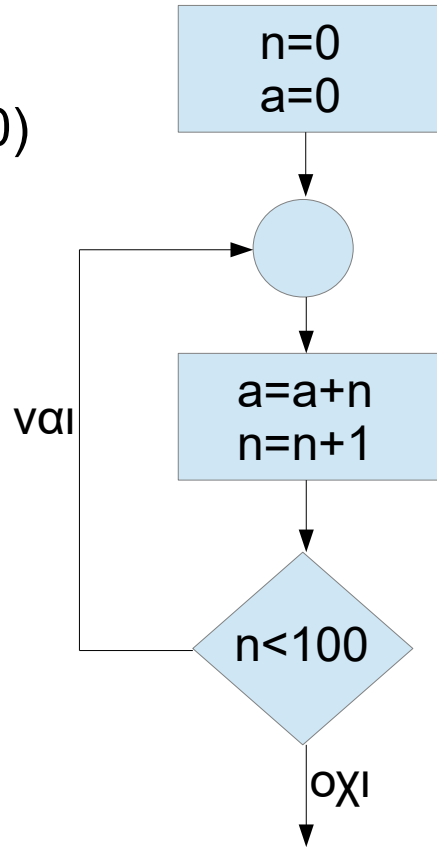
- Διακλαδώσεις προς τα πίσω (νέα διεύθυνση < παρούσα διεύθυνση). Διακλαδώσεις αυτού του τύπου είναι συνήθως οι βρόγχοι (loops)
- Εξαιρέσεις (Exceptions) και διαχείριση λαθών (error handling)
- Παράδειγμα στατικής πρόγνωσης
 - Διακλάδωση προς τα πίσω : Θα γίνει
 - Διακλάδωση προς τα εμπρός : Δεν θα γίνει

Ορισμένοι επεξεργαστές επιτρέπουν την εισαγωγή *υποδείξεων* στο κώδικα για το πού πρέπει να γίνει ή να μή γίνει διακλάδωση. Σε αυτή τη περίπτωση είναι δυνατό να χρησιμοποιηθούν πληροφορίες από το ιστορικό της εκτέλεσης (profile) του προγράμματος

```

n = 0;
a = 0;
while (n < 100)
{
  a = a + n;
  n = n + 1;
}

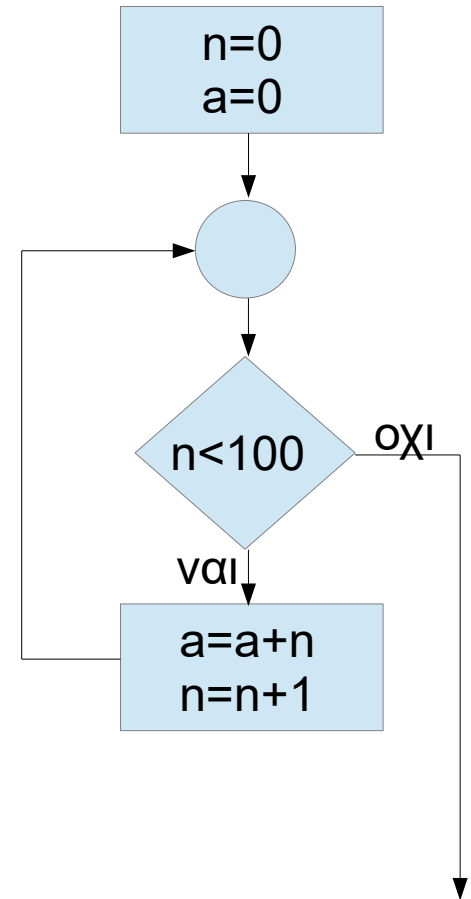
```



```

n = 0;
a = 0;
do
{
  a = a + n;
  n = n + 1;
}
while (n < 100)

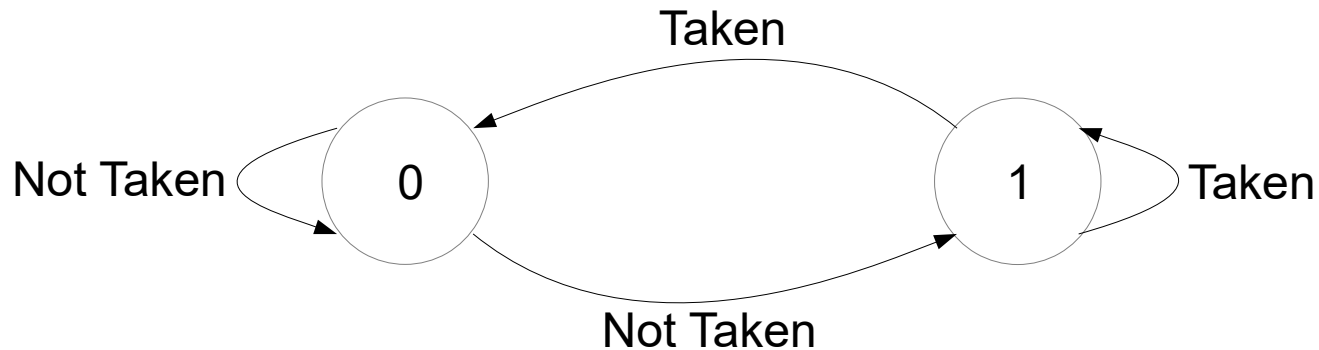
```



Δυναμική πρόβλεψη

Μια περιοχή μνήμης (buffer) περιέχει το αποτέλεσμα των προηγούμενων διακλαδώσεων με βάση τα οποία θα γίνει ή πρόβλεψη.

- Πρόβλεψη με 1 bit
 - '1' : Θα γίνει
 - '0' : Δεν θα γίνει



Η πρόβλεψη με ένα bit μπορεί να προκαλέσει δύο εσφαλμένες προβλέψεις σε βρόγχους επανάληψης (loops)

Στην αρχή του βρόγχου : Η πρόβλεψη θα είναι “Not Taken” αλλά θα γίνει διακλάδωση

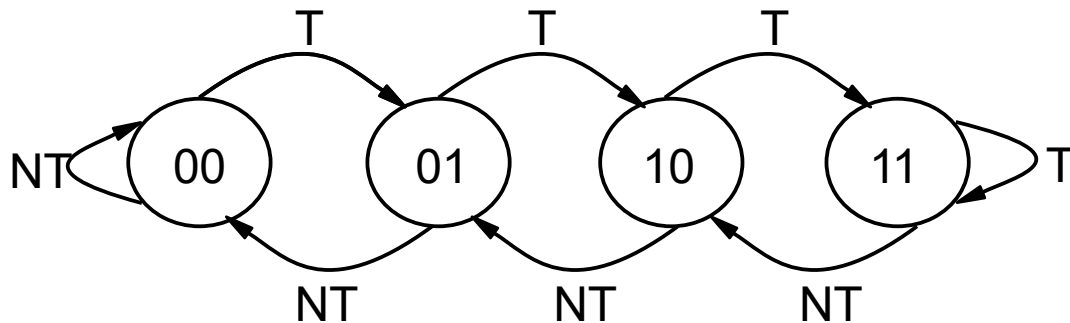
Στο τέλος του βρόγχου : Η πρόβλεψη είναι “Taken” αλλά δεν θα γίνει διακλάδωση.

Αποτελέσματα
Προβλέψεις

1111011110111101
111**10**111**10**111**10**

Μετρητής κορεσμού Saturating counter

Μηχανή καταστάσεων (state machine)



Καταστάσεις(Πρόβλεψη)

00 : Strongly Not Taken

01 : Weakly Not Taken

10 : Weakly Taken

11 : Strongly Taken

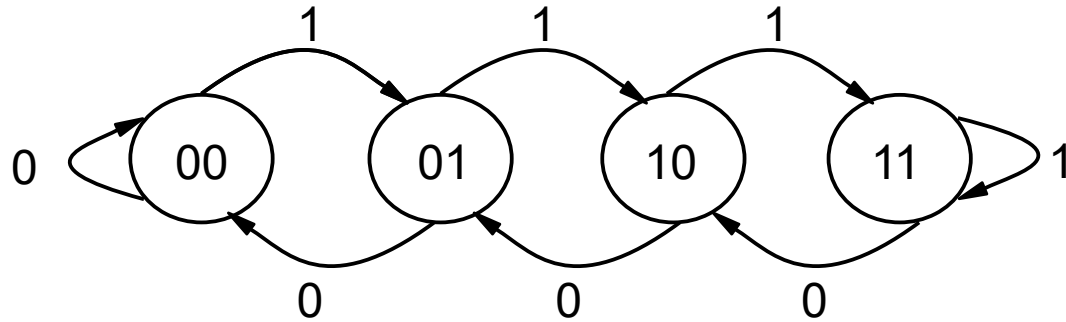
Μεταβάσεις:

T(1) : έγινε (Taken)

NT(0): δεν έγινε (Not Taken)

- Έγινε διακλάδωση (branch taken) νέα κατ.=παρούσα κατ. -1
 - Η κατάσταση “11” παραμένει “11” μετά από αύξηση (κόρος)
- Δεν έγινε διακλάδωση (branch not taken) νέα κατ.=παρούσα κατ.-1
 - Η κατάσταση “00” παραμένει “00” μετά από μείωση (κόρος)
- Στις καταστάσεις “11” και “10” η πρόβλεψη είναι “taken” (θα γίνει)
- Στις καταστάσεις “00” και “01” η πρόβλεψη είναι “not-taken” (δεν θα γίνει)

Παραδειγμα



Αρχική κατάσταση 11 (δεκ. 3)

Αποτελέσματα

1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1

Καταστάσεις

3 3 3 3 2 3 3 3 3 2 3 3 3 3 2

Προβλέψεις

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Συσχέτιση προβλέψεων

- Συστήματα πρόβλεψης των 2 bits χρησιμοποιούν την πρόσφατη συμπεριφορά μιας εντολής διακλάδωσης για να προβλέψουν τη μελλοντική συμπεριφορά της.
- Η συμπεριφορά μεγαλύτερης ακολουθίας από το ιστορικό της διακλάδωσης μπορεί να δώσει καλύτερες προβλέψεις
- Υπάρχουν περιπτώσεις που η συμπεριφορά και άλλων εντολών διακλάδωσης επηρεάζουν το αποτέλεσμα.
 - Τα αποτελέσματα διαφόρων διακλαδώσεων συσχετίζονται
 - Καθολικό ιστορικό διακλαδώσεων
- Για ορισμένες διακλαδώσεις το προηγούμενο ιστορικό είναι σημαντικό
 - Σε βρόγχους Because of loops
 - Τοπικό ιστορικό Local branch history

Συσχετισμός Correlation

```
B1 : if (x == 1)
      y = 2;
      else
        y = 5;
```

```
B3 : if (z == 1)
      w = 2;
      else
        w = 7;
```

```
B3 : if ( y == w)
```

.....

Η διακλάδωση B3 μπορεί να προβλεφθεί με ακρίβεια 100% με δεδομένα τα αποτελέσματα των B1 και B2

Τοπικό σύστημα πρόβλεψης Local Predictor

- Ένα τοπικό σύστημα πρόβλεψης έχει μία προσωρινή μνήμη ιστορικού (history buffer) για κάθε εντολή πρόβλεψης άλματος
- Ο πίνακας με το πρότυπο του ιστορικού μπορεί να είναι διαφορετικός για κάθε άλμα ή κοινός για όλα τα άλματα υπό συνθήκη
- Η ακρίβεια του συστήματος είναι 97.1%

Καθολικό σύστημα πρόβλεψης Global Predictor

- Καθολική πρόβλεψη
- Ένα σύστημα καθολικής πρόβλεψης διατηρεί κοινό ιστορικό όλων των αλμάτων υπό συνθήκη. Κάθε συσχέτιση μεταξύ διαφορετικών αλμάτων υπό συνθήκη αξιοποιείται στην πρόβλεψη, αλλά όταν δεν υπάρχει συσχέτιση ο βαθμός πρόβλεψης μπορεί ελαττωθεί.
- Το σύστημα πρόβλεψης πρέπει να έχει μεγάλη προσωρινή μνήμη ιστορικού. Το μέγεθος του πίνακα προτύπου του ιστορικού αυξάνεται εκθετικά με το μέγεθος της προσωρινής μνήμης ιστορικού. Έτσι ο πίνακας είναι κοινός για όλα τα άλματα υπό συνθήκη.
- Η ακρίβεια του συστήματος καθολικής πρόβλεψης είναι 96.6%

Στη πράξη χρησιμοποιείται ένας συνδυασμός των δύο συστημάτων πρόβλεψης.

Branch target prediction

- Η διεύθυνση μιας **δυναμικής** διακλάδωσης δεν είναι γνωστή κατά τη μεταγλώττιση, αλλά υπολογίζεται κατά την εκτέλεση του προγράμματος.
- Branch target predictor. είναι το τμήμα του επεξεργαστή που προβλέπει τη διεύθυνση στόχο μιας δυναμικής υπό συνθήκη ή χωρίς συνθήκη εντολής διακλάδωσης πριν από τον υπολογισμό από την μονάδα εκτέλεσης εντολών της διεύθυνσης στόχου.

Απλή πρόβλεψη διακλάδωσης

Κώδικας assembly:

```
branch if condition to label1
  do_that
  branch to label2
label1:
  do_this
label2:
  ....
```

Προβλεπόμενος κώδικας:

```
(condition) do_this
(not condition) do_that
```

Και οι δύο κλάδοι του προγράμματος εκτελούνται και μετά τη διακλάδωση ο κλάδος που δεν χρειάζεται απορρίπτεται.

Πλεονεκτήματα : Ταχύτερη εκτέλεση αν οι κλάδοι έχουν μικρό αριθμό εντολών.
Μειονεκτήματα : Χρησιμοποιεί περισσότερη μνήμη. Μη ισορροπημένοι κλάδοι.

Complex Instruction Set Computer Reduced Instruction Set Computer

RISC (PowerPC)

- Έμφαση στο software
- Σχετικά μικρός αριθμός απλών εντολών που εκτελούνται ταχύτερα (1 εντολή ανά κύκλο ρολογιού).
- Ομοιόμορφες εντολές που αποκωδικοποιούνται ευκολότερα.
- Απλούστεροι τρόποι διευθυνσιοδότησης
- Μεγάλο μέγεθος κώδικα
- Πολλοί καταχωρητές γενικής χρήσης

CISC (Pentium)

- Έμφαση στο hardware
- Σύνθετες εντολές (Συνήθως απαιτούνται πολλοί κύκλοι ρολογιού)
- Μικρός κώδικας
- Πολλοί σύνθετοι τρόποι διευθυνσιοδότησης
- Λίγοι καταχωρητές

Οι δύο τεχνολογίες συγκλίνουν. Το αποτέλεσμα είναι οι επεξεργαστές να προσδιορίζονται ως RISC ή CISC μόνο από την καταγωγή τους.

Intel® 64 and IA-32 Architectures
Software Developer's Manual

325462-050US

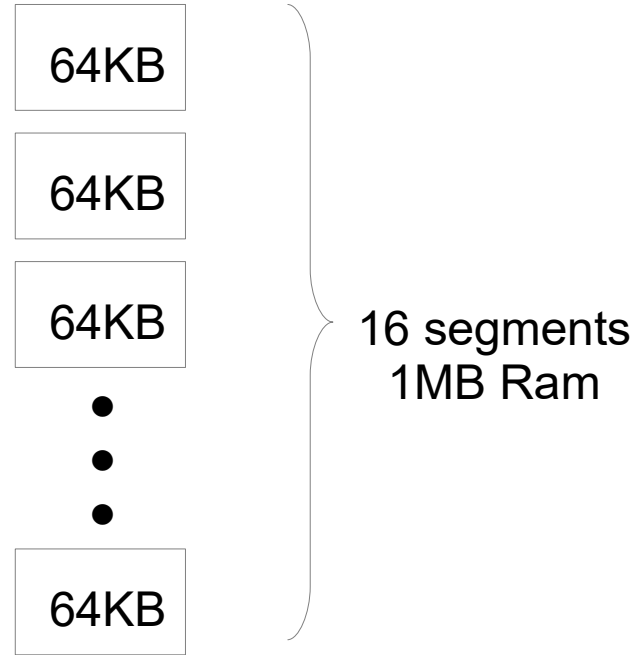
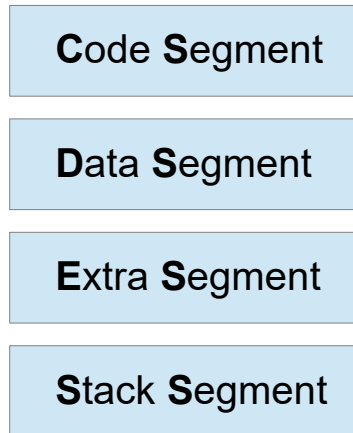
Volume 1:
Basic Architecture

253665-050US

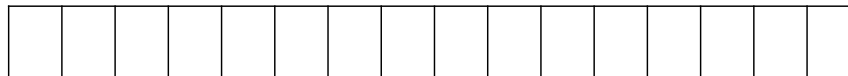
Segments

16 bit (8086)

Segment Registers

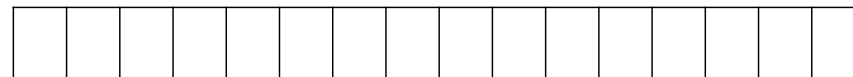


Segment Register



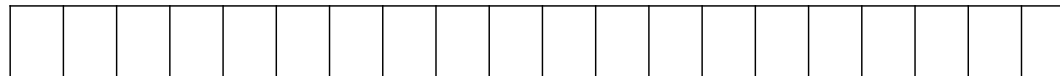
16bit

+ Offset



16bit

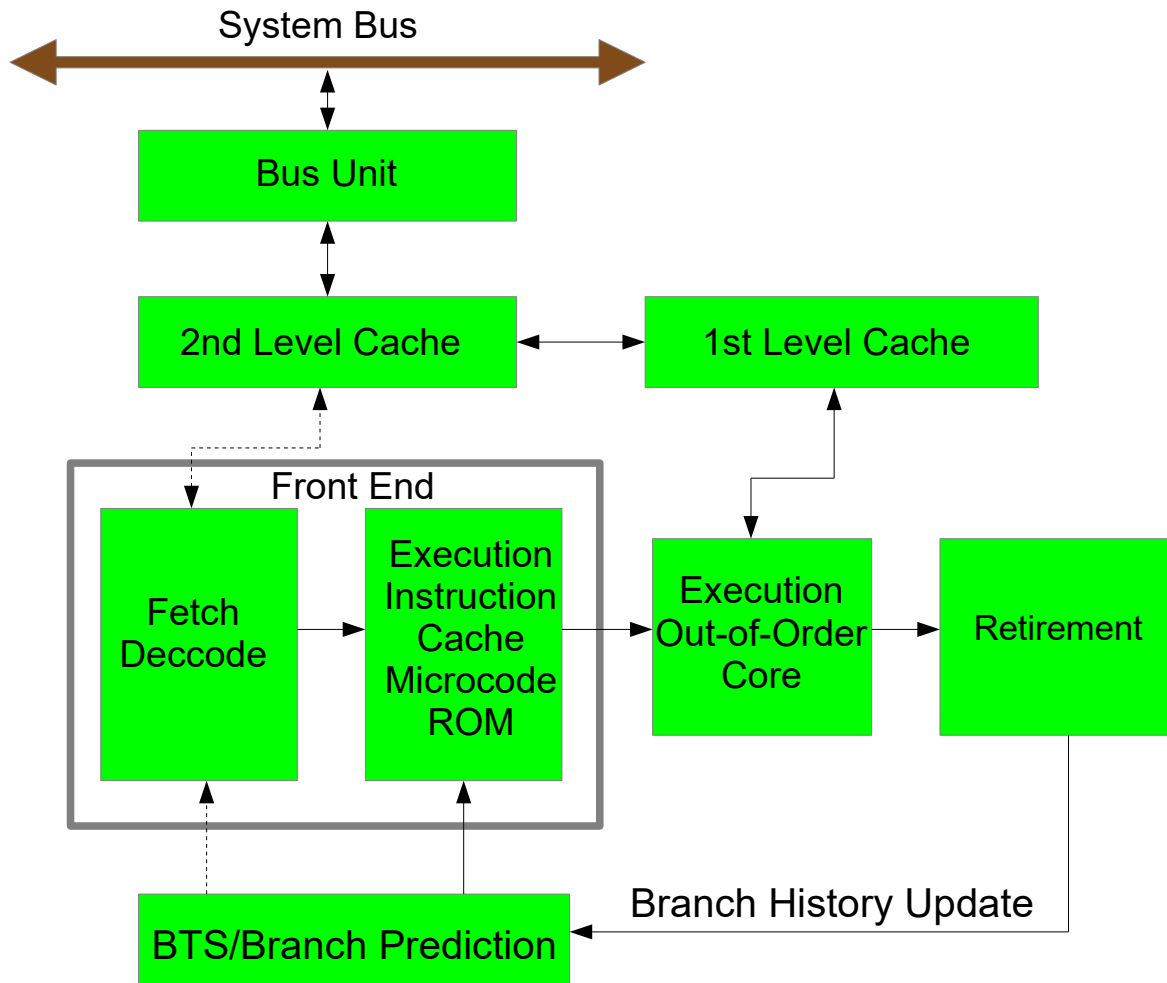
Memory Address

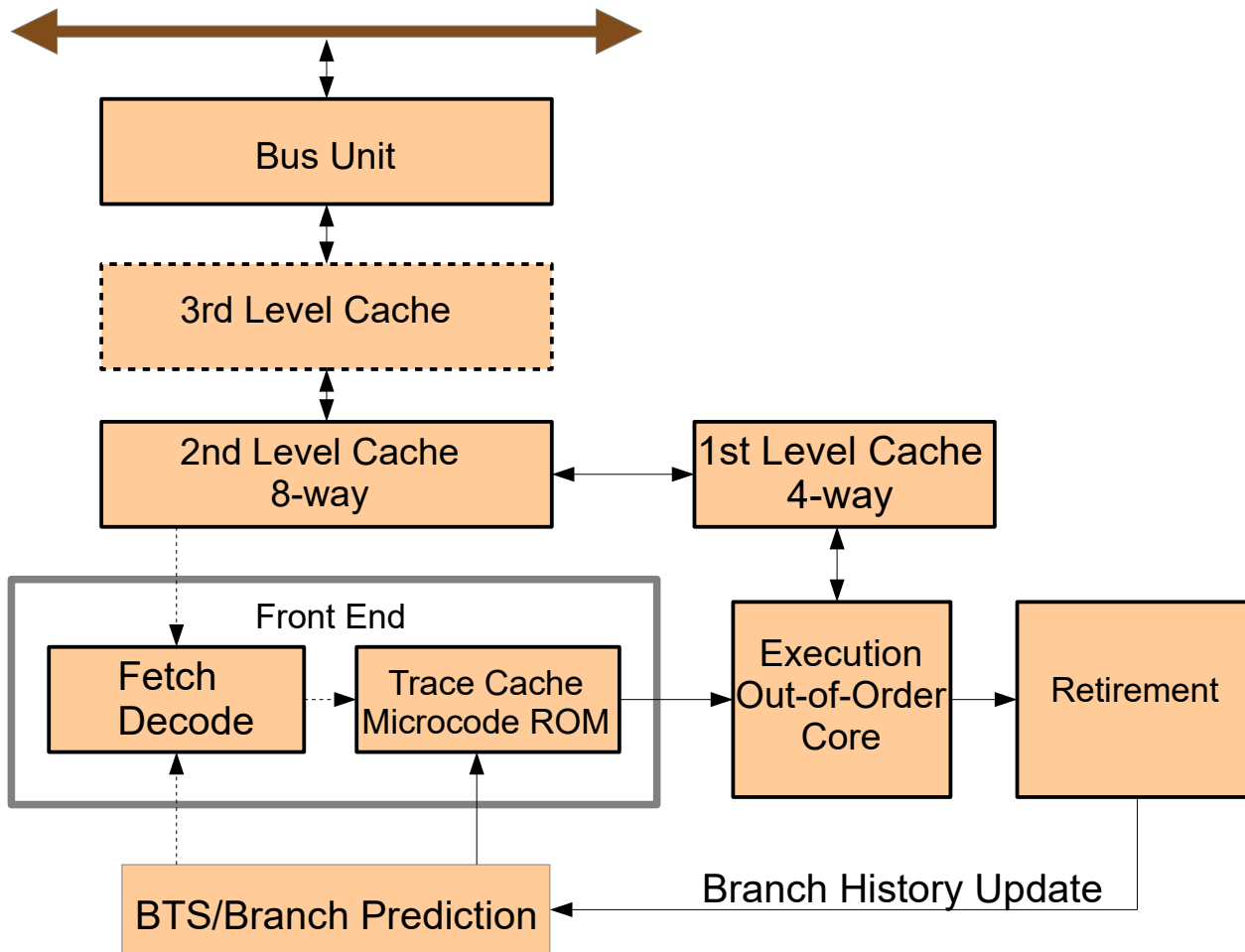


20bit

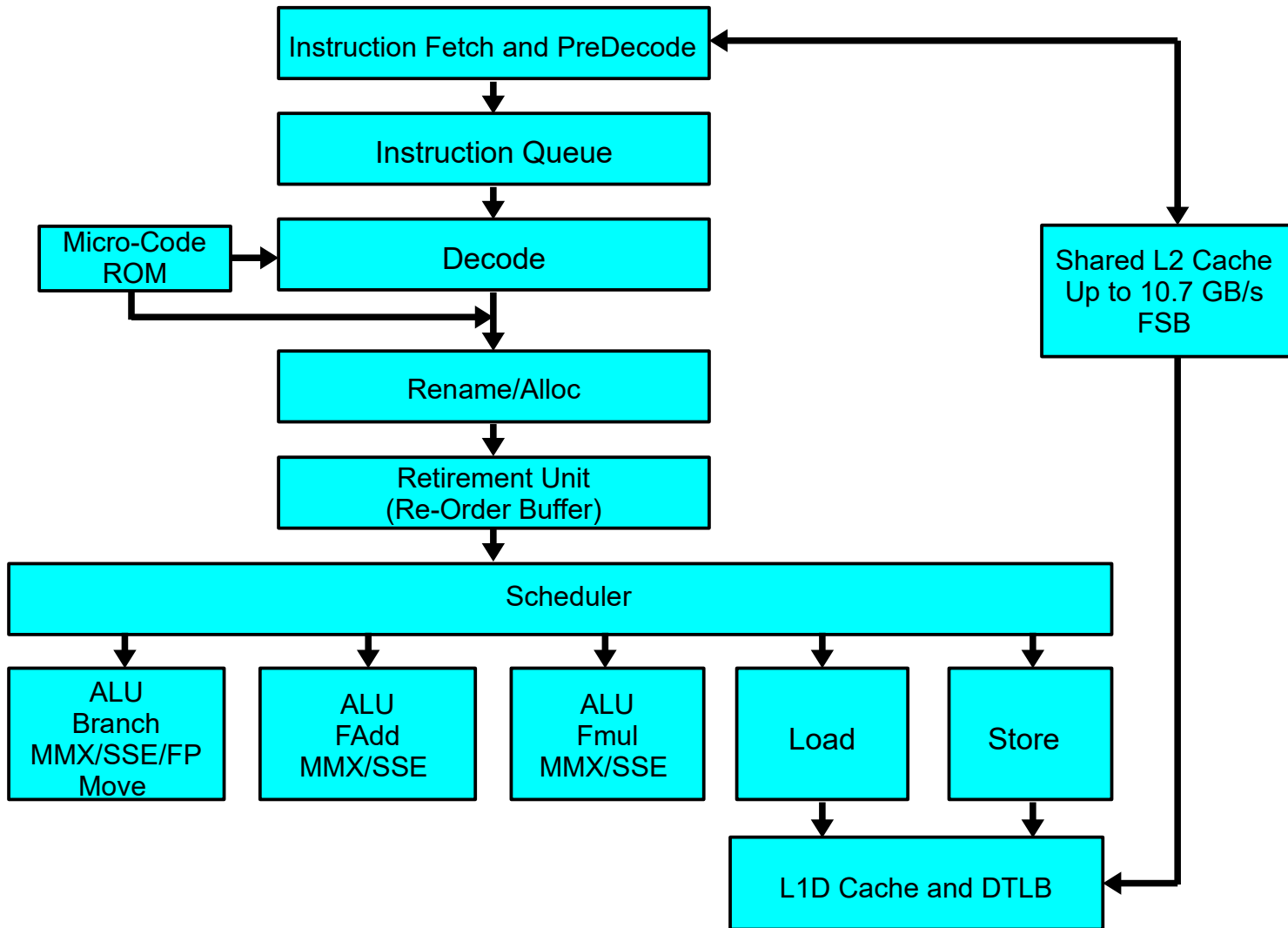
- 80286
 - 16 bit data ,24 bit address (16MB Ram)
 - Segments 64KB , limit checking
 - Privilege levels
 - Read-only , Execute-only

- 80386
 - 32 bit data & address (4GB Ram)
 - A segmented-memory model and a flat memory model
 - Paging, with a fixed 4-KByte page size providing a method for virtual memory management
 - Support for parallel stages





Επεξεργαστές αρχιτεκτονικής Core



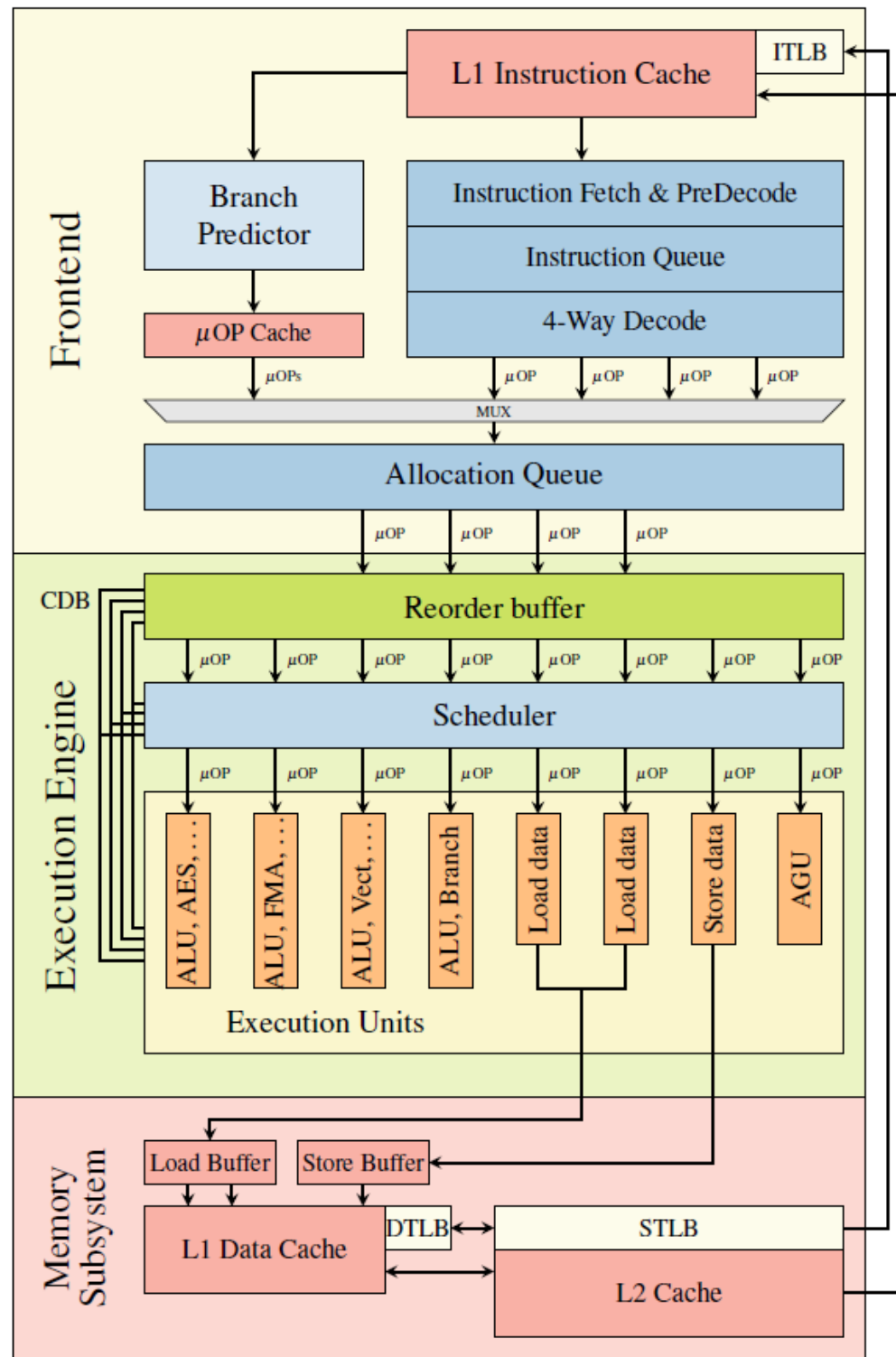
- **Out-Of-Order Execution.** Αλλαγή της σειράς εκτέλεσης των μικροεντολών, όταν δεν αλλοιώνονται τα δεδομένα (integrity), ώστε να αξιοποιούνται οι δυνατότητες του επεξεργαστή για παραλληλία. Απαιτείται ανάλυση για να εντοπιστούν εξαρτήσεις των δεδομένων (Dynamic data flow analysis).
- **Retirement Unit.** Λαμβάνει τα δεδομένα που προέκυψαν από την εκτός σειράς εκτέλεση μικροεντολών και τα επεξεργάζεται έτσι ώστε να είναι σύμφωνα με την αρχική ροή του προγράμματος. Όταν μια μικροεντολή εκτελεστεί *αποσύρεται*
 - **The Reorder Buffer (ROB).** Συγκρατεί τις μικροεντολές που ολοκληρώθηκαν, διαχειρίζεται τις εξαιρέσεις, παρακολουθεί τις διακλαδώσεις και ενημερώνει τον Branch Target Buffer για το προορισμό των διακλαδώσεων. Ο BTB εκκαθαρίζει τις εντολές που αποκωδικοποιήθηκαν αλλά είναι άχρηστες.
- **Speculative execution.** Η δυνατότητα του επεξεργαστή να εκτελεί μικροεντολές που βρίσκονται μετά από μια διακλάδωση υπό συνθήκη. Τα αποτελέσματα αποθηκεύονται σε προσωρινούς καταχωρητές .

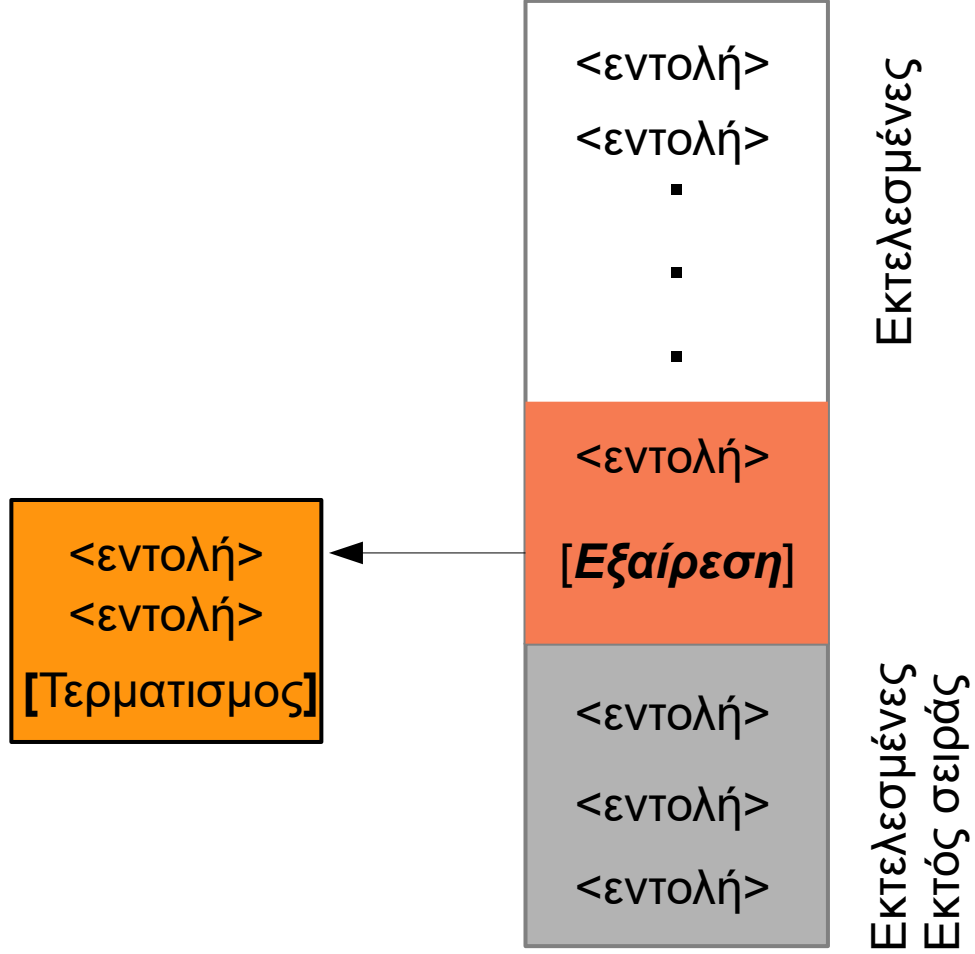
- **The Front End**

- Μεταφέρει τις επόμενες εντολές από τη μνήμη (PreFetch) και τις τοποθετεί σε μια ουρά (instruction queue) ώστε να υπάρχει συνεχής ροή προς τη μονάδα αποκωδικοποίησης (decode)
- Η ουρά εντολών μπορεί να κρατήσει σε κρυφή μνήμη (caching) σύντομους βρόγχους για αυξημένη απόδοση.
- Πρόβλεψη διακλαδώσεων

- **Execution Core**

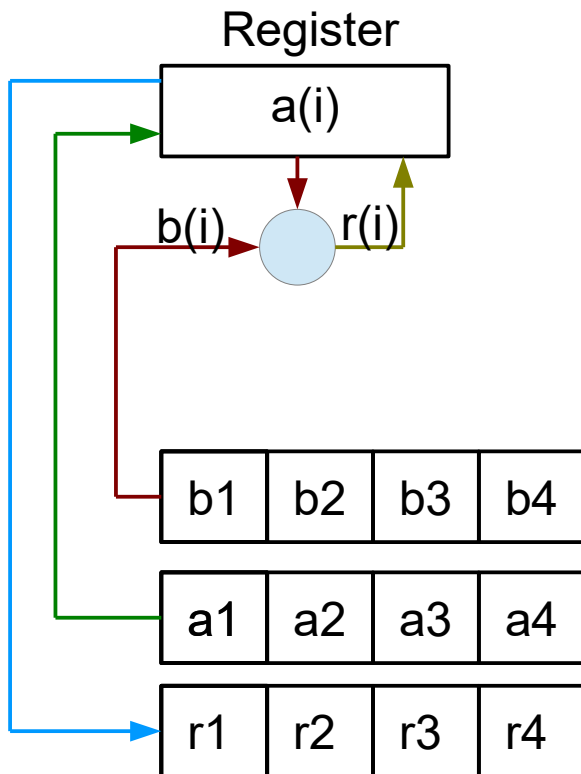
- Δυνατότητα εκτέλεσης 6 μικροεντολών ανά κύκλο μηχανής (clock cycle)
- Εως 4 μικροεντολές μπορούν να αποσυρθούν σε κάθε κύκλο.
- Τρεις Αριθμητικές Μονάδες (ALU)
- Εως 8 πράξεις κινητής υποδιαστολής ανά κύκλο





Single Instruction Single Data Single Instruction Multiple Data

SISD



SIMD

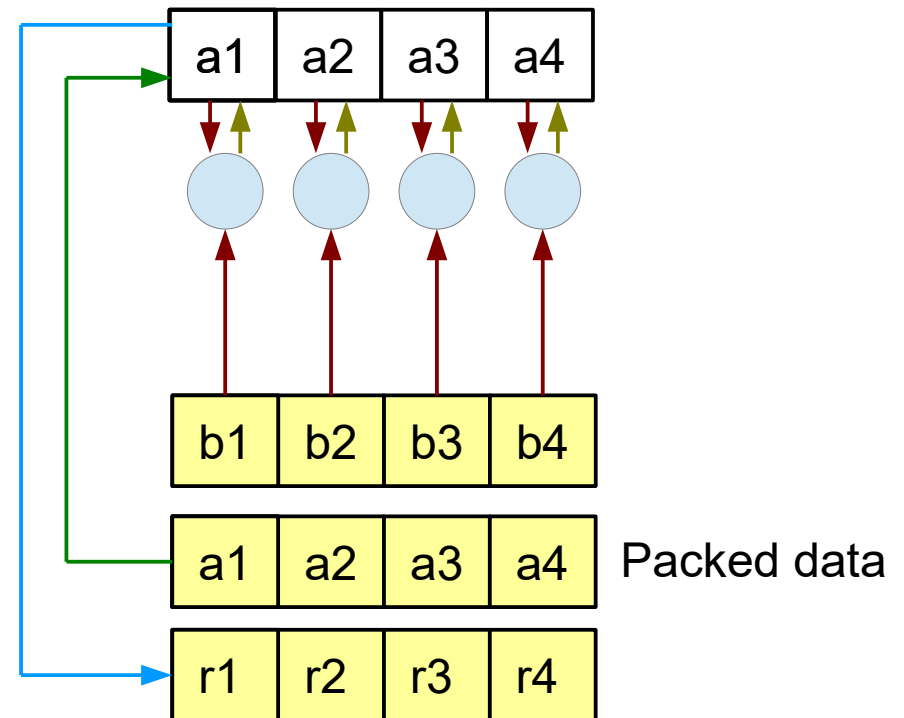
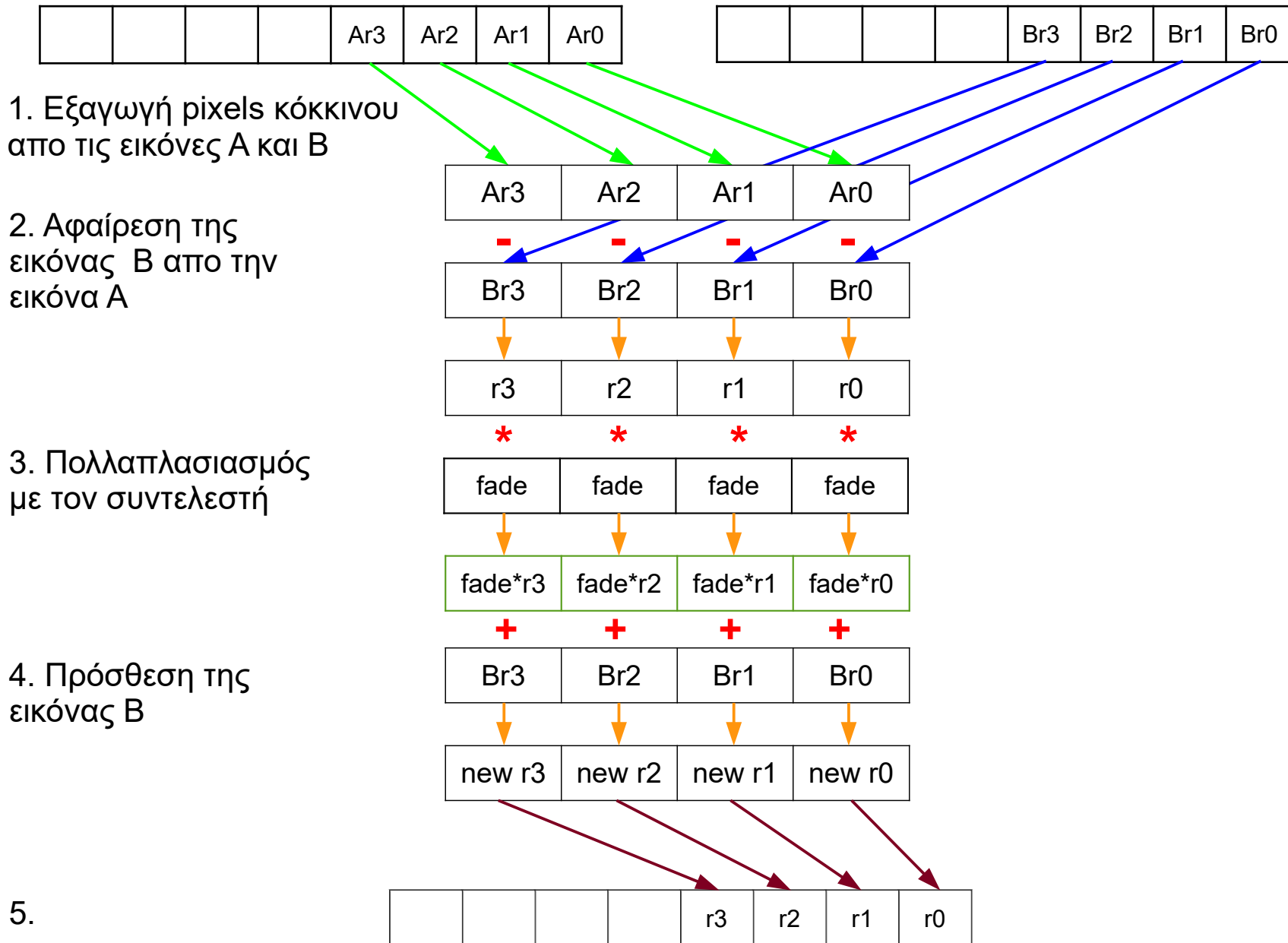


image fade-in-fade-out

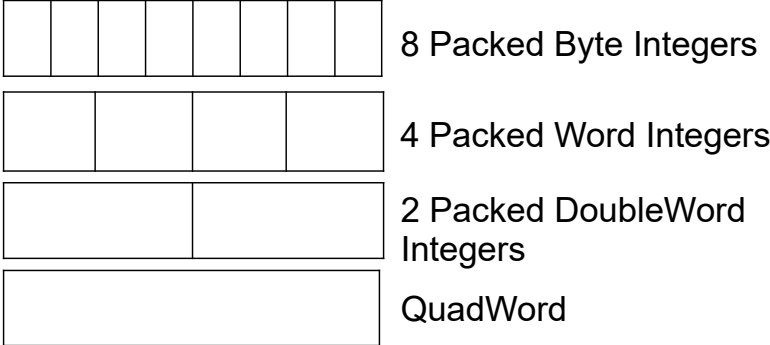


SIMD

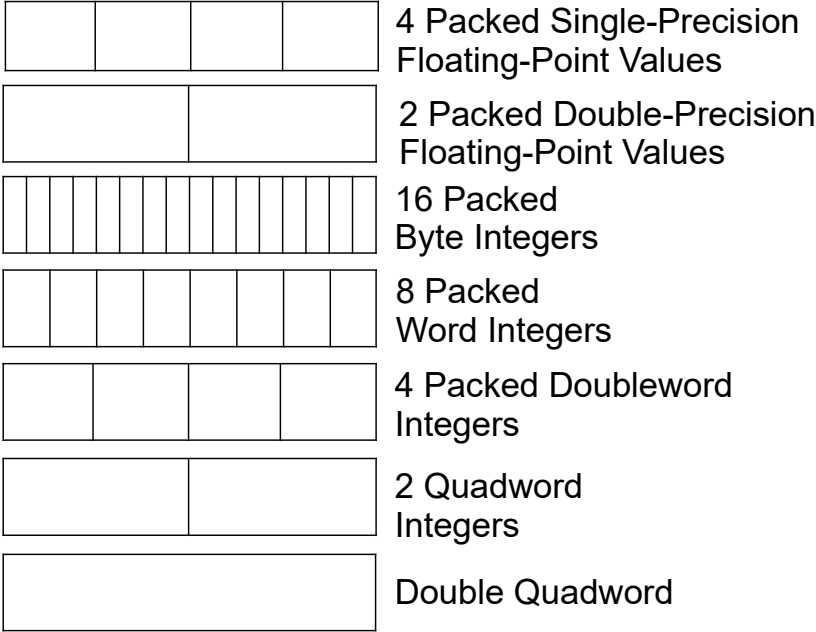
- Εντολές MMX (Multi Media eXtensions , Matrix Math eXtension)
- Για τα δεδομένα υπάρχουν οι καταχωρητές MMX (64bits) για ακεραίους και XMM (128bits) για πραγματικούς.
- SSE (Streaming Simd eXtensions) packed single-precision floating-point, arrays
- SSE2 extensions double-precision floating-point
- SSE3 integers
- SSSE3 integers
- SSE4
- AESNI and PCLMULQDQ AES encryption/decryption standard
- AVX (Advanced Vector eXtensions)

Registers

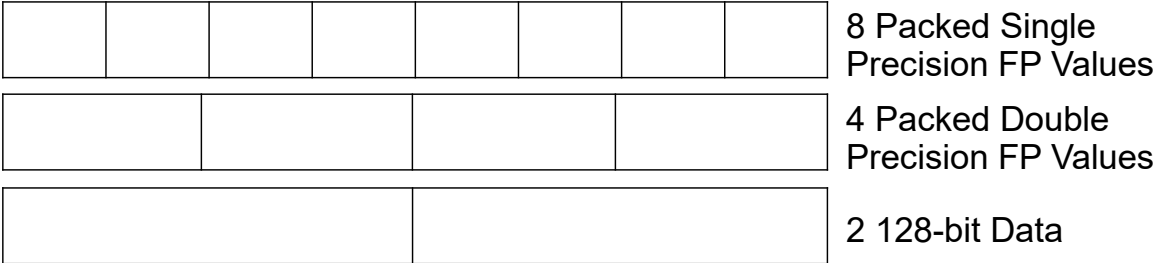
MMX Registers (64BIT) MMX-SSSE3



XMM Registers (128bit) SSE - AVX

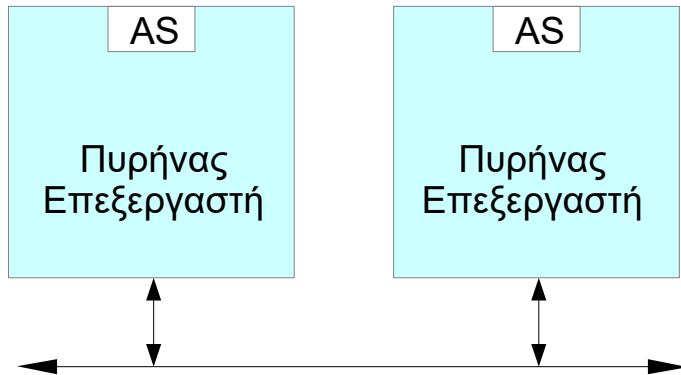


YMM Registers (256bit) AVX

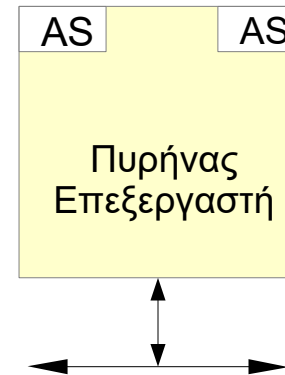


Hyper-Threading

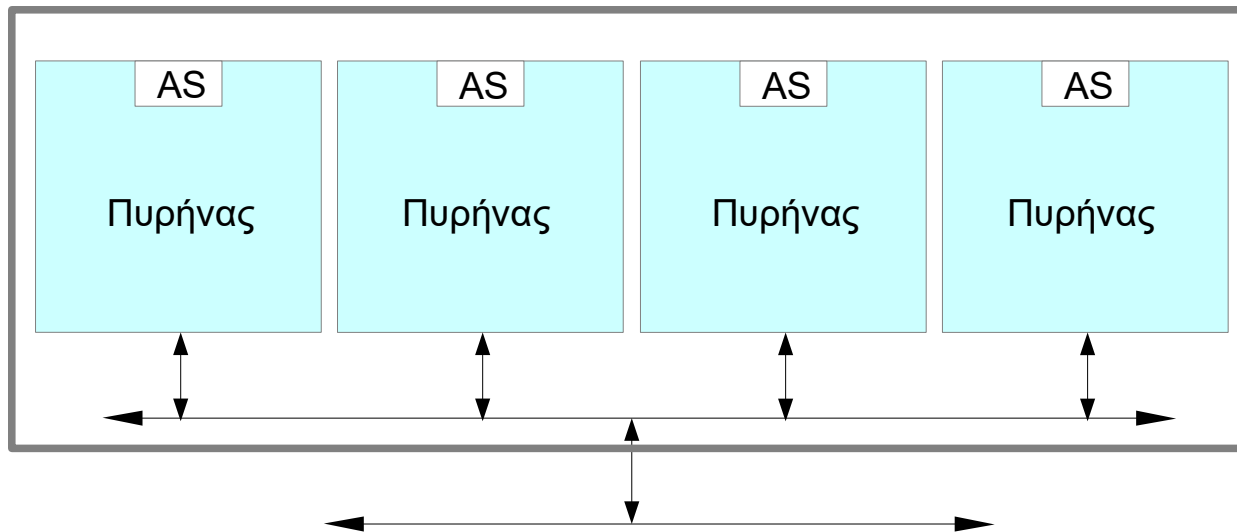
Δύο επεξεργαστές
Μια συσκευασία



Δύο πλήρη σύνολα καταχωρητών
Μια μονάδα εκτέλεσης εντολών



Νομος του Moore
ο αριθμός των
τρανζίστορ που θα μπορούσε να
ενσωματωθεί σε μια μήτρα πυριτίου
θα διπλασιάζεται κάθε 18 μήνες για
τα επόμενα χρόνια



Modes of Operation 32bit

- **Protected mode**

- Η κανονική κατάσταση λειτουργίας.
- Όλες οι δυνατότητες του επεξεργαστή ενεργές.
- Virtual 8086 mode

- **Real-address mode**

- 8086 1MB Ram
- πρόσθετες δυνατότητες
- Η κατάσταση αμέσως μετά το reset.

- **System management mode (SMM)**

- Ο επεξεργαστής έρχεται σε αυτή τη κατάσταση μετά από ένα σήμα διακοπής SMM
- Λειτουργίες διαχείρισης ισχύος και ασφάλειας
- Χωριστή περιοχή μνήμης

Modes of Operation σε 64bit

Compatibility mode (sub-mode of IA-32e mode)

Εφαρμογές 16bit και 32bit σε λειτουργικό σύστημα 64bit χωρίς νέ μεταγλώττιση

Εφαρμογές 64bit

Ομοιο με το 32-bit protected mode

Μέγιστη μνήμη 4GBytes ή 64GBytes με PAE

64-bit mode (sub-mode of IA-32e mode)

64-bit εφαρμογές με πρόσβαση σε όλη τη μνήμη

8 επιπλέον καταχωρητές γενικής χρήσης

Πρόθεμα REX για πρόσβαση στους καταχωρητές με επέκταση

Registers IA-32

I/O Ports

Control registers

Memory Management registers

Debug Registers

Memory type range registers (MTRRs)

Machine specific registers (MSRs)

Machine check registers

Performance monitoring counters

X87 FPU Registers

MMX Registers

XMM Registers

BASIC EXECUTION ENVIRONMENT 32-bit

Basic Program Execution Registers

8 καταχωρητές 32-bit	General Purpose Registers
6 καταχωρητές 16-bit	Segment Registers
32-bit	EFLAGS Register
32-bit	EIP Register



FPU Registers

IEEE

8 καταχωρητές 80-bit	Floating Point Registers
16-bit	Control Register
16-bit	Status Register
16-bit	Tag Register
11-bit	Opcode Register
48-bit	FPU IP register
48-bit	FPU Data Pointer Register

MMX Registers

8 καταχωρητές 64-bit

XMM Registers

8 καταχωρητές 128-bit	
32-bit	MXCSR

64-bit

Basic Program Execution Registers

16 καταχωρητές 64-bit	General Purpose Registers (R1-R8)
6 καταχωρητές 16-bit	Segment Registers
64-bit	EFLAGS Register
64-bit	EIP Register



FPU Registers

8 καταχωρητές 80-bit	Floating Point Registers
16-bit	Control Register
16-bit	Status Register
16-bit	Tag Register
11-bit	Opcode Register
64-bit	FPU IP register
64-bit	FPU Data Pointer Register

MMX Registers

8 καταχωρητές 64-bit

XMM Registers

8 καταχωρητές 128-bit	
32-bit	MXCSR

Οργάνωση μνήμης

- **Linear address space.** Η περιοχή διευθύνσεων που καθορίζεται από το μήκος του καταχωρητή IP (32 ή 64 bit) $2^{32}-1$ bytes ή $2^{64}-1$ bytes
- **Physical memory.** Η μνήμη που μπορεί να συνδεθεί στον επεξεργαστή εξαρτάται από τις διαθέσιμες γραμμές του διαύλου διευθύνσεων (address bus)
 - Επεξεργαστές 32bit : $2^{32}-1$ bytes (4 Gbytes)
 - Επεξεργαστές 32bit Physical Address Extension $2^{36}-1$ bytes (64 Gbytes)
 - Επεξεργαστές 64bit : $2^{40}-1$ bytes
 -

Η μνήμη οργανώνεται σε τμήματα (segments) και σελίδες (pages) για αποδοτική διαχείριση και ασφάλεια.

Μοντέλα μνήμης 32-bit

- **Flat memory model.**

- Μια ενιαία και συνεχής περιοχή μνήμης που ονομάζεται γραμμική περιοχή διευθύνσεων (linear address space)
- Linear address : Η διεύθυνση ενός byte

- **Segmented memory model**

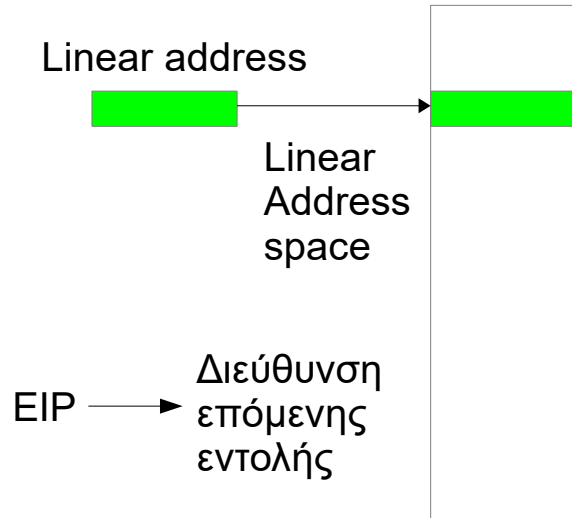
- Η μνήμη οργανώνεται σε ένα σύνολο ανεξάρτητων περιοχών που ονομάζονται τμήματα (segments)
- Χωριστά τμήματα για κώδικα προγράμματος (Code Segment) , δεδομένα (Data Segment) και στοίβα (Stack Segment)
- Logical Address : Η διεύθυνση ενός byte μέσα σε ένα segment. (Segment + offset)

- **Real-address mode memory model.**

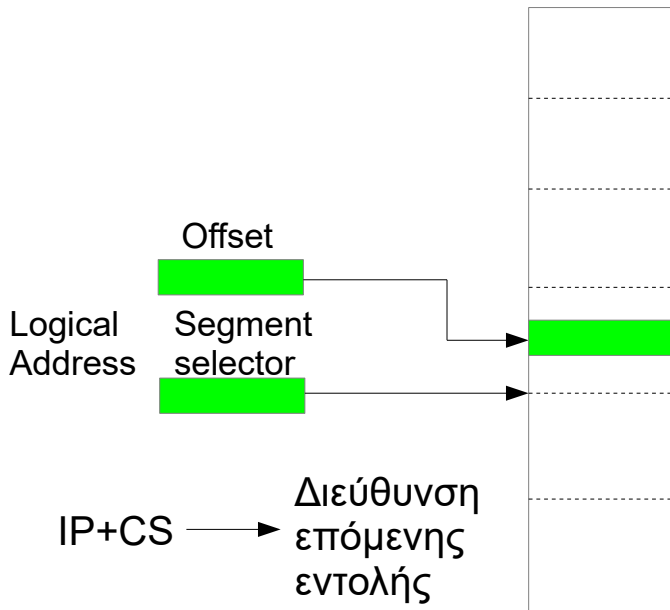
- Το μοντέλο μνήμης του 8086. (1 Mbyte Physical Memory)
- Προσφέρει συμβατότητα για προγράμματα γραμμένα για 8086.

Memory models IA-32

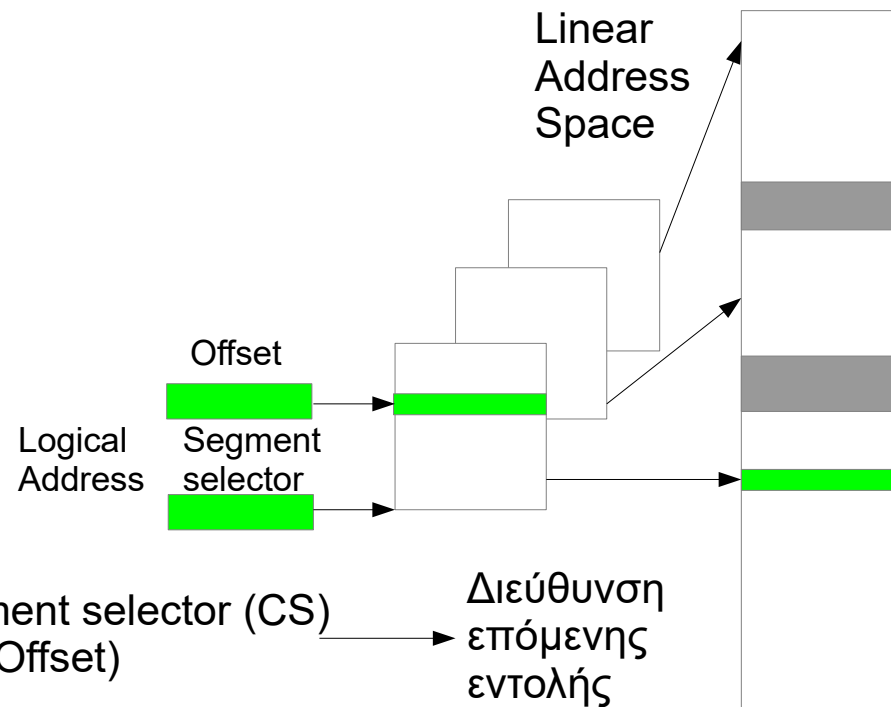
Flat model



Real-Address Mode Model



Segmented Model



Memory Models 64bit

- Linear address space $2^{64}-1$
- Physical address space. Εξαρτάται από τον επεξεργαστή

Τρόποι λειτουργίας και μοντέλα μνήμης

Modes of Operation vs Memory Model

- **Protected mode**
 - Μπορούν να χρησιμοποιηθούν όλα τα μοντέλα μνήμης (Εξαρτάται από το λειτουργικό σύστημα).
 - Κάθε διεργασία μπορεί να χρησιμοποιεί διαφορετικό μοντέλο
 - Virtual 8086 mode
- **Real-address mode**
 - Real-Address Memory Model.
- **System management mode (SMM)**
 - Χωριστή περιοχή μνήμης (SMRAM)
 - Οργάνωση μνήμης όμοια με το Real-Address Memory Model.
- **Compatibility Mode.**
 - Μοντέλο μνήμης ίδιο με το 32-bit Protected mode
- **64-bit mode.**
 - Δεν χρησιμοποιούνται segments εκτός από ειδικές περιπτώσεις (Οι αρχικές διευθύνσεις των CS,DS,ES,SS θεωρούνται μηδέν)
 - Δεν υπάρχουν segmented και real-address modes

Διευθύνσεις και δεδομένα σε 16 και 32 bit

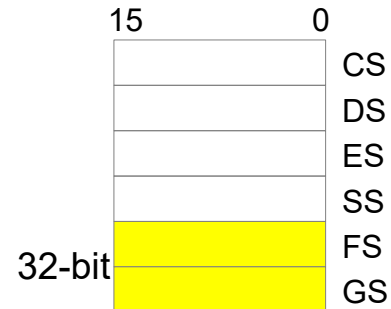
- 16 bit mode
 - Διευθύνσεις 16 bit (16bit segment selector+ 16 bit offset)
 - Δεδομένα 8 ή 16 bit
- 32 bit mode
 - Διευθύνσεις 32 bit (16bit segment selector + 32 bit offset, Far pointer)
 - Δεδομένα 32 , 16 , 8 bit
- Extended Physical Addressing (32bit mode)
 - Διευθύνσεις 36bit. 16 περιοχές 4GBytes
- 64 bit mode.
 - Η αρχική διεύθυνση των CS,DS,ES,SS είναι μηδέν
 - FS,GS. Η αρχική διεύθυνση μπορεί να μην είναι μηδέν

BASIC PROGRAM EXECUTION REGISTERS 16 και 32-bit

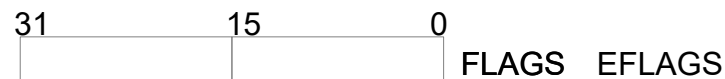
Καταχωρητές γενικής χρήσης

31	15	7	0	16-bit	32-bit
	AH	AL		AX	EAX
	BH	BL		BX	EBX
	CH	CL		CX	ECX
	DH	DL		DX	EDX
	BP				EBP
	SI				ESI
	DI				EDI
	SP				ESP

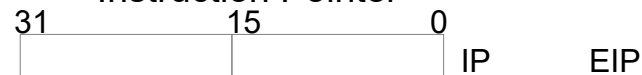
Segment Registers



Status-Control Register



Instruction Pointer



Καταχωρητές γενικής χρήσης

General-Purpose Registers

- Δεδομένα (τελεστέοι-Operands) για αριθμητικές και λογικές πράξεις
 - Operands για υολογισμό διεθύνσεων
 - Δείκτες μνήμης (Memory pointers)
-
- EAX : Accumulator (δεδομένα, αποτελέσματα)
 - EBX : Δείκτης στο Data Segment
 - ECX : Μετρητής για εντολές επαναλήψεων
 - EDX : Δείκτης στη περιοχή I/O
 - ESI : Δείκτης στο DS (Source Index, string operations)
 - EDI : Δείκτης στο ES (Destination index, string operations)
 - EBP : Δείκτης στο SS (Base Pointer)
 - ESP : Stack Pointer

```

mov esi, source_address    ; pointer in DS (IA-32)
mov edi, target_address    ; pointer in ES (IA-32)
mov ecx, number_of_words
cld                        ; set direction forward
rep movsw                  ; repeat until cx=0

```

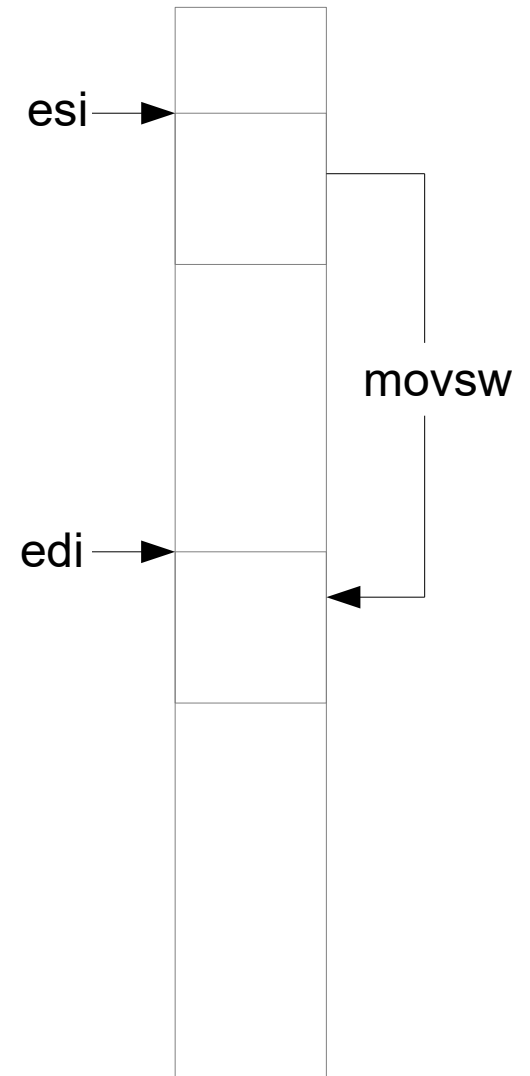
Μετά την εκτέλεση της movsw

df=0	df=1
esi=esi+1	esi=esi-1
edi=edi+1	edi=edi-1

```

n=number_of_words
while(n>=0)
{
  s1++ = s2++;
  n--;
}

```



General-Purpose Registers 64-bit

8-bit	AL,BL,CL,DL,AH,BH, CH,DH	AL,BL,CL,DL,DIL,SIL, BPL,SPL,R8L-R15L
16-bit	AX,BX,CX,DX,DI,SI, BP,SP	AX,BX,CX,DX,DI,SI,BP, SP,R8W-R15W
32-bit	EAX,EBX,ECX,EDX, EDI,ESI,EBP,ESP	EAX,EBX,ECX,EDX,EDI, ESI,EBP,ESP,R8D-R15D
64bit		RAX,RBX,RCX,RDX,RDI, RSI,RBP,RSP,R8-R15

Segment Registers 32-bit

Οι καταχωρητές CS,DS,ES,SS,FS,GS περιέχουν 16-bit δείκτες που ονομάζονται segment selectors.

- Η χρήση τους εξαρτάται από το μοντέλο μνήμης
- Flat model. Όλοι οι καταχωρητές περιέχουν segment selectors που είναι δείκτες στη διεύθυνση 0
- Segmented model. Κάθε καταχωρητής μπορεί να έχει διαφορετικό segment selector. Το πρόγραμμα σε κάθε στιγμή μπορεί να έχει πρόσβαση σε 6 segments
- CS : εντολές (CS:EIP)
- DS,ES,FS,GS : Δεδομένα
- SS : Stack (SS:ESP)

Segment Registers 64-bit

- CS,DS,ES,SS : Ανεξάρτητα από τη διεύθυνση που δείχνει ο selector ως αρχική διεύθυνση λαμβάνεται η 0
- FS,GS. Μπορούν να χρησιμοποιηθούν για δεδομένα και για δομές του λειτουργικού συστήματος
- Ένας segment register μπορεί να χρησιμοποιηθεί σε λειτουργία 64-bit από μία εφαρμογή που τρέχει σε λειτουργία συμβατότητας.

Καταχωρητής κατάστασης Program status and control register EFLAGS register

- Bits που δείχνουν το αποτέλεσμα μιας πράξης
 - Κρατούμενο (Carry Flag): Μια πρόσθεση έχει κρατούμενο
 - Μηδέν (Zero Flag): Η αριθμητική ή λογική πράξη έδωσε αποτέλεσμα μηδέν
 - Υπερχείλιση (Overflow Flag)
- Bits που ρυθμίζουν τη λειτουργία του επεξεργαστή
 - Interrupt Flag : Επιτρέπεται ή δεν επιτρέπεται ή εξυπηρέτηση διακοπών
 - Virtual-8086 mode flag : Ενεργοποίηση (0) , επιστροφή (1) από τη λειτουργία του 8086 mode

Instruction Pointer

EIP (32 bit) RIP (64 bit)

(Program Counter)

- Περιέχει τη διεύθυνση (CS offset) της επόμενης εντολής
- Η τιμή δεν αλλάζει άμεσα με εντολή αλλά ελέγχεται εμμέσως από εντολές ελέγχου ροής του προγράμματος (διακλαδώσεις)

Operand Addressing

(Addressing Modes, Τρόποι διευθυνσιοδότησης)

- Ενας τελεστέος μπορεί να υπάρχει
 - Στον κώδικα της εντολής : `ADD EAX,10` (άμεσα δεδομένα, immediate)
Εκτός πολλαπλασιασμού και διαίρεσης
 - Σε ένα καταχωρητή : `ADD EAX,EBX`
 - Στη μνήμη
 - Σε I/O Port
- Τα αποτελέσματα υπάρχουν
 - Σε ένα καταχωρητή (ή δύο για πολλαπλασιασμό)
 - Στη μνήμη
 - Σε I/O Port
-

- Τελεστέοι μνήμης
 - Segment selector 16bits : offset (linear address) 32 ή 16bits
 - Segment selector 16bits : offset (linear address) 64bits
- Εντολή MOV :Για μεταφορά δεδομένων από μνήμη χρησιμοποιείται πάντα το DS εκτός αν προσδιορίζεται άλλο.
 - MOV EAX,*address*
 - MOV EAX,[ECX]
 - MOV EAX,ES:[ECX]
- 64 bit Mode

Προσδιορισμός offset effective address

$$\begin{pmatrix} \text{EAX} \\ \text{EBX} \\ \text{ECX} \\ \text{EDX} \\ \text{ESP} \\ \text{EBP} \\ \text{ESI} \\ \text{EDI} \end{pmatrix} + \left[\begin{pmatrix} \text{EAX} \\ \text{EBX} \\ \text{ECX} \\ \text{EDX} \\ \text{EBP} \\ \text{ESI} \\ \text{EDI} \end{pmatrix} * \begin{pmatrix} 1 \\ 2 \\ 4 \\ 8 \end{pmatrix} \right] + \begin{pmatrix} \text{NONE} \\ 8\text{-bit} \\ 16\text{-bit} \\ 32\text{-bit} \end{pmatrix}$$

$$\text{Offset} = \text{Base} + (\text{Index} * \text{Scale}) + \text{Displacement}$$

Τα δεδομένα είναι στο DS εκτός εάν καταχωρητής βάση (base) είναι ο ESP ή ο EBP που είναι το SS

- Displacement : Η άμεση (direct) διεύθυνση ή στατική διεύθυνση offset.
- Base : Έμμεση διεύθυνση offset
- Base + Displacement : Πρόσβαση σε στοιχεία πίνακα.
 - Displacement = Διεύθυνση πρώτου στοιχείου
 - Base = διεύθυνση του ν-οστού στοιχείου
- (Index * Scale) + Displacement : Πίνακας με στοιχεία 2,4,8 bytes
- Base + Index + Displacement : Πίνακας δύο διαστάσεων
- Base + (Index * scale) + Displacement : Πίνακας δύο διαστάσεων με στοιχεία 2,4,8 bytes

integer a[10],k;

..

..

k = *(a + 5) *(a + 5*n)

Προσδιορισμός offset 64bit

- **Displacement** : 8-bit, 16-bit, 32-bit
- **Base** : 32-bit (or 64-bit if REX.W is set) καταχωρητής.
- **Index** : Η τιμή ενός καταχωρητή γενικής χρήσης 32-bit (ή 64-bit if REX.W is set).
- **Scale factor** : Συντελεστής 2, 4, ή 8