

# Απόδοση συσκευών Ε/Ε

---

## Κριτήρια απόδοσης σύνθετα

- **access latency** - Πόσο χρόνο χρειάζεται για να ξεκινήσει η μεταφορά δεδομένων – μετράται σε χρόνο  
*«Για να παίξουμε Quake 3 θέλουμε όσο το δυνατόν μικρότερο latency»*

# Απόδοση συσκευών Ε/Ε

---

## Κριτήρια απόδοσης σύνθετα

- **access latency** - Πόσο χρόνο χρειάζεται για να ξεκινήσει η μεταφορά δεδομένων – μετράται σε χρόνο  
*«Για να παίξουμε Quake 3 θέλουμε όσο το δυνατόν μικρότερο latency»*
- **throughput** – Διεκπεραιωτική ικανότητα, πόσο γρήγορα μεταφέρονται δεδομένα στη μονάδα του χρόνου – μετράται σε bytes/sec  
*«Για να κατεβάσουμε μια ταινία, θέλουμε όσο το δυνατόν μεγαλύτερο throughput»*

# Απόδοση συσκευών Ε/Ε

---

Η απόδοση μιας συσκευής Ε/Ε εξαρτάται από:

- χαρακτηριστικά της συσκευής
- σύνδεση συσκευής με υπόλοιπο σύστημα
- ιεραρχία μνήμης
- λειτουργικό σύστημα

# Οργάνωση συσκευών Ε/Ε

---

Βάσει των ακόλουθων χαρακτηριστικών:

- **Συμπεριφορά (behavior)**
  - read only, write only, read/write
- **Εταίρος (partner)**
  - άνθρωπος ή μηχανή τροφοδοτεί δεδομένα
- **Ρυθμός δεδομένων (data rate)**
  - μέγιστος ρυθμός μεταφοράς δεδομένων μεταξύ συσκευής και μνήμης ή CPU

# Οργάνωση συσκευών Ε/Ε

---

Βάσει των ακόλουθων χαρακτηριστικών:

- **Συμπεριφορά (behavior)**
  - read only, write only, read/write
- **Εταίρος (partner)**
  - άνθρωπος ή μηχανή τροφοδοτεί δεδομένα
- **Ρυθμός δεδομένων (data rate)**
  - μέγιστος ρυθμός μεταφοράς δεδομένων μεταξύ συσκευής και μνήμης ή CPU

Π.χ. πληκτρολόγιο – συσκευή *read only*,  
χρησιμοποιείται από *άνθρωπο*, με *data rate 10 bytes/sec*

# Οργάνωση συσκευών Ε/Ε

Συσκευή	Συμπεριφορά	Εταίρος	Ρυθ Δεδομένων (Mbit/sec)
Πληκτρολόγιο	Input	Άνθρωπος	0,0001
Ποντίκι	Input	Άνθρωπος	0,0038
Είσοδος Φωνής	Input	Άνθρωπος	0,2640
Είσοδος Ήχου	Input	Μηχανή	3
Scanner	Input	Άνθρωπος	3,2
Έξοδος φωνής	Output	Άνθρωπος	0,2640
Έξοδος ήχου	Output	Άνθρωπος	8
Laser printer	Output	Άνθρωπος	3,2
Οθόνη γραφικών	Output	Άνθρωπος	800 – 8000,
Modem	Input/Output	Μηχανή	0,0160– 0,064
Network/LAN	Input/Output	Μηχανή	100 – 1000
Network/Wireless	Input/Output	Μηχανή	11 – 54
Optical Disk	Input/Output	Μηχανή	80
Magnetic Tape	Input/Output	Μηχανή	32
Magnetic Disk	Input/Output	Μηχανή	240 - 2560

# Τρόπος Αξιολόγησης Απόδοσης Ε/Ε

---

Εξαρτάται από την εφαρμογή

Σε μερικά περιβάλλοντα μας νοιάζει το **throughput**  
Τότε το **bandwidth** είναι το πιο σημαντικό.

Μετράται με 2 τρόπους:

- Data που διακινούμε τη μονάδα του χρόνου  
ή
- Λειτουργίες Ε/Ε που πραγματοποιούμε τη μονάδα του χρόνου

# Τρόπος Αξιολόγησης Απόδοσης Ε/Ε

---

Σε άλλες εφαρμογές μας νοιάζει ο χρόνος απόκρισης  
(response time)

Εξαρτάται από **bandwidth** και **access latency**

Για μεγάλες αιτήσεις Ε/Ε



Για μικρές αιτήσεις Ε/Ε





# Τρόπος Αξιολόγησης Απόδοσης Ε/Ε

---

Πλήθος εφαρμογών απαιτούν  
υψηλή διεκπεραιωτική ικανότητα  
και  
μικρό χρόνο απόκρισης

- ATM
- File servers
- Web servers
- κ.α.

*«Μας ενδιαφέρει και η διάρκεια κάθε εργασίας αλλά και πόσες διεργασίες εκτελούνται το δευτερόλεπτο»*

# Σύνοψη Απόδοσης E/E

---

Desktop, Servers, Embedded Systems  
φερεγγυότητα + κόστος E/E

Desktop , Embedded Systems  
χρόνος απόκρισης + ποικιλία συσκευών E/E

Servers  
διεκπεραιωτική ικανότητα συσκευών E/E

# Αποθήκευση στο δίσκο – φερεγγυότητα

---

Αποθήκευση σε δίσκο μη πτητική (non-volatile)

*«Τα δεδομένα παραμένουν και όταν διακόπτεται η τροφοδοσία»*

# Αποθήκευση στο δίσκο – φερεγγυότητα

---

Δίσκος αποτελείται:

- **Στοίβα πλακών** (1 – 4) με 2 επιφάνειες εγγραφής
- Στοίβα πλακών **περιστρέφεται** (5400 – 15000 RPM)
- Κάθε επιφάνεια διαιρείται σε ομόκεντρους κύκλους – **τροχιές/tracks** (10K – 50K tracks/επιφάνεια)
- Κάθε τροχιά διαιρείται σε **τομείς/sectors** (100 – 500 sectors/track)
- Τυπικό μέγεθος τομέα **512bytes** (με τάση προς τα 4KB)

# Προσπέλαση Δεδομένων στο Δίσκο

---

α) **Seek time** (τοποθέτηση κεφαλής πάνω από κατάλληλη τροχιά)

β) **Rotational Latency** ή **rotational delay** (χρόνος για να βρεθεί κεφαλή πάνω από κατάλληλο τομέα)

γ) **Transfer Time** (χρόνος που απαιτείται για τη μεταφορά ενός block από bits)

# Πλεονασματικές Συστοιχίες Φθηνών Δίσκων - RAID

---

Redundant Array of Inexpensive Disks

*Διάταξη δίσκων που χρησιμοποιεί μια συστοιχία “μικρών” και “φθηνών” δίσκων ώστε να αυξήσει τόσο την **απόδοση** όσο και την **αξιοπιστία**.*

RAID: “Πολλαπλασιάζει” τα read “heads”, redundancy χρειάζεται γιατί οι “μικροί” & “φθηνοί” δεν είναι αξιόπιστοι.

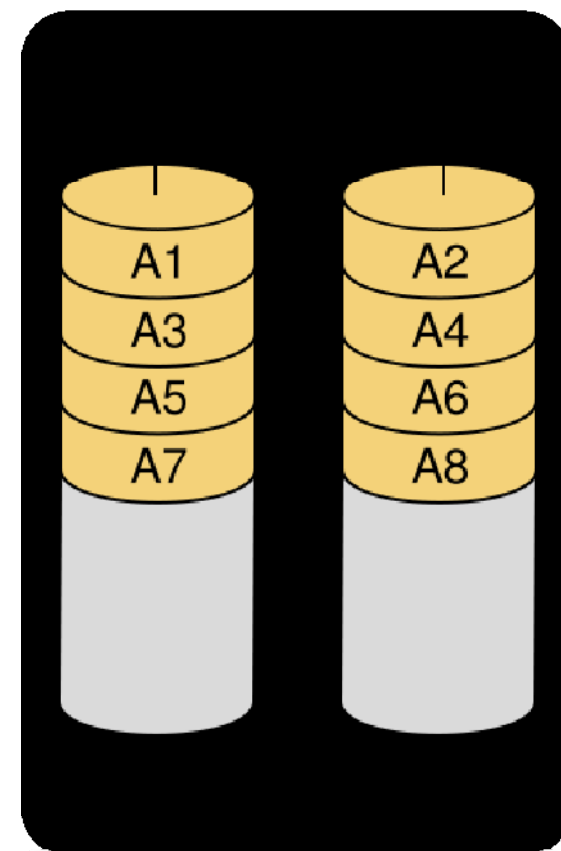
# RAID 0 - Striping

Συνεχόμενα blocks γράφονται στους 2 δίσκους εναλλάξ.

Παρέχει υψηλή απόδοση αφού οι εγγραφές και οι αναγνώσεις γίνονται συνεχώς σε διαφορετικούς δίσκους «διπλασιάζοντας» τις επιδόσεις.

Όμως δεν παρέχει αξιοπιστία, αφού η βλάβη ενός δίσκου καταστρέφει ολόκληρη τη συστοιχία.

(Ο όρος RAID στην περίπτωση αυτή χρησιμοποιείται καταχρηστικά)



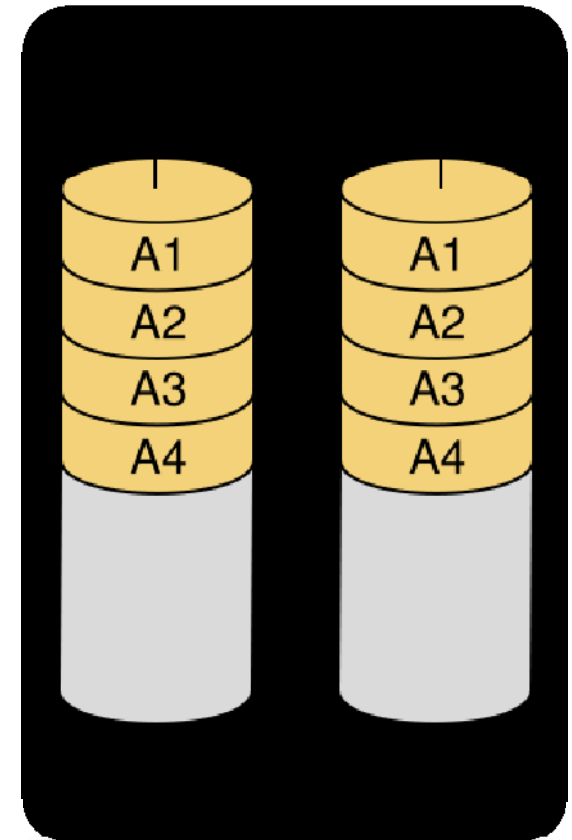
# RAID 1 - Mirroring

---

Κάθε block γράφεται και στους 2 δίσκους (αντίγραφο).

Παρέχει υψηλή απόδοση στις αναγνώσεις, αφού αυτές μπορούν να γίνουν από 2 δίσκους εναλλάξ.

Παρέχει αξιοπιστία, αφού αν πάθει βλάβη ένας δίσκος, τα δεδομένα υπάρχουν στον 2<sup>ο</sup>.





- 
- Έχουμε όγκο δεδομένων που χωράει σε 4 δίσκους
  - Αγοράζουμε 8 φυσικούς δίσκους
  - Πώς θα τους οργανώσουμε για mirroring και striping;

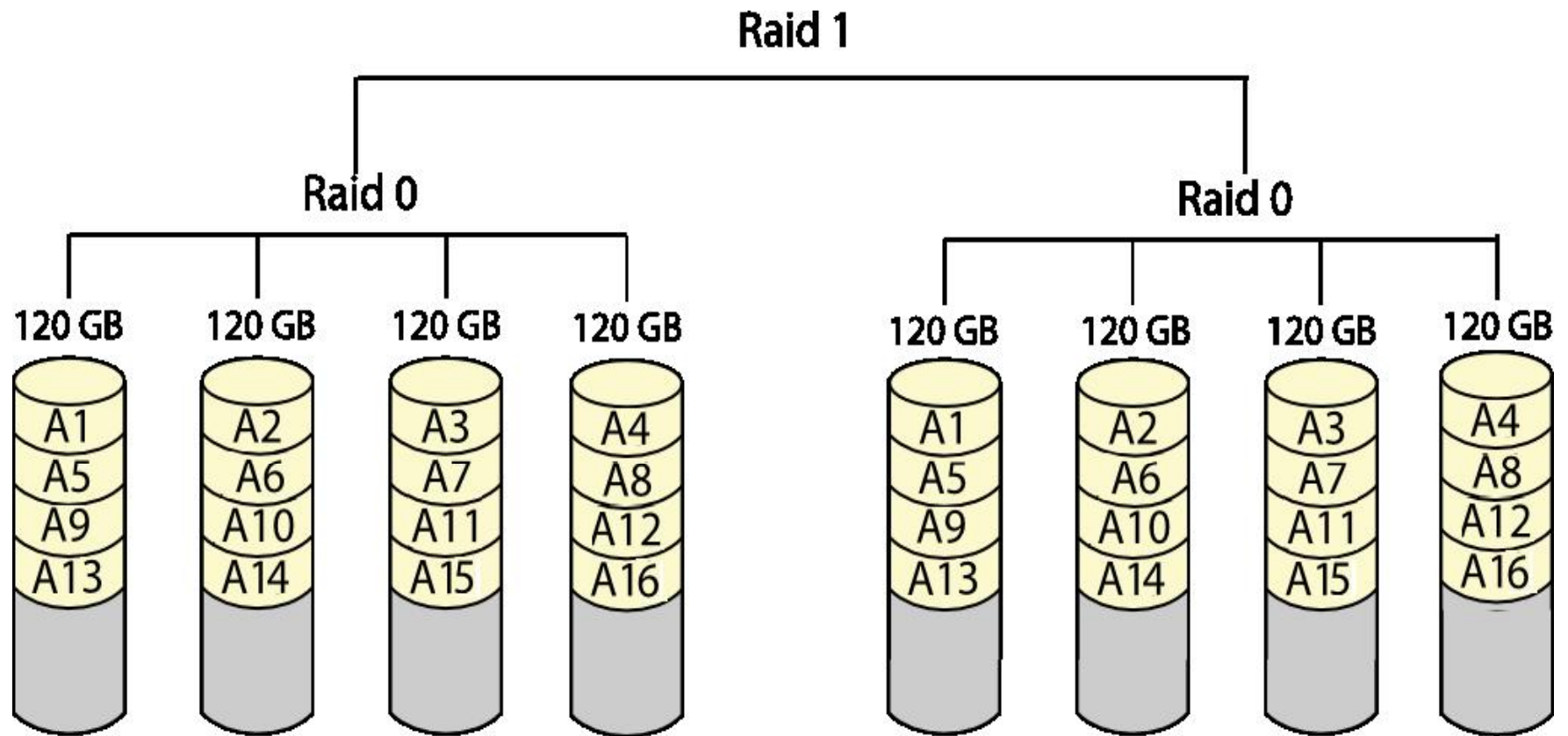
## RAID 0+1 (RAID 01)

Φτιάχνουμε 2 σύνολα των 4 δίσκων, το κάθε σύνολο το οργανώνουμε σε RAID 0 (striping) και τα 2 σύνολα είναι mirror το ένα του άλλου (RAID-1)

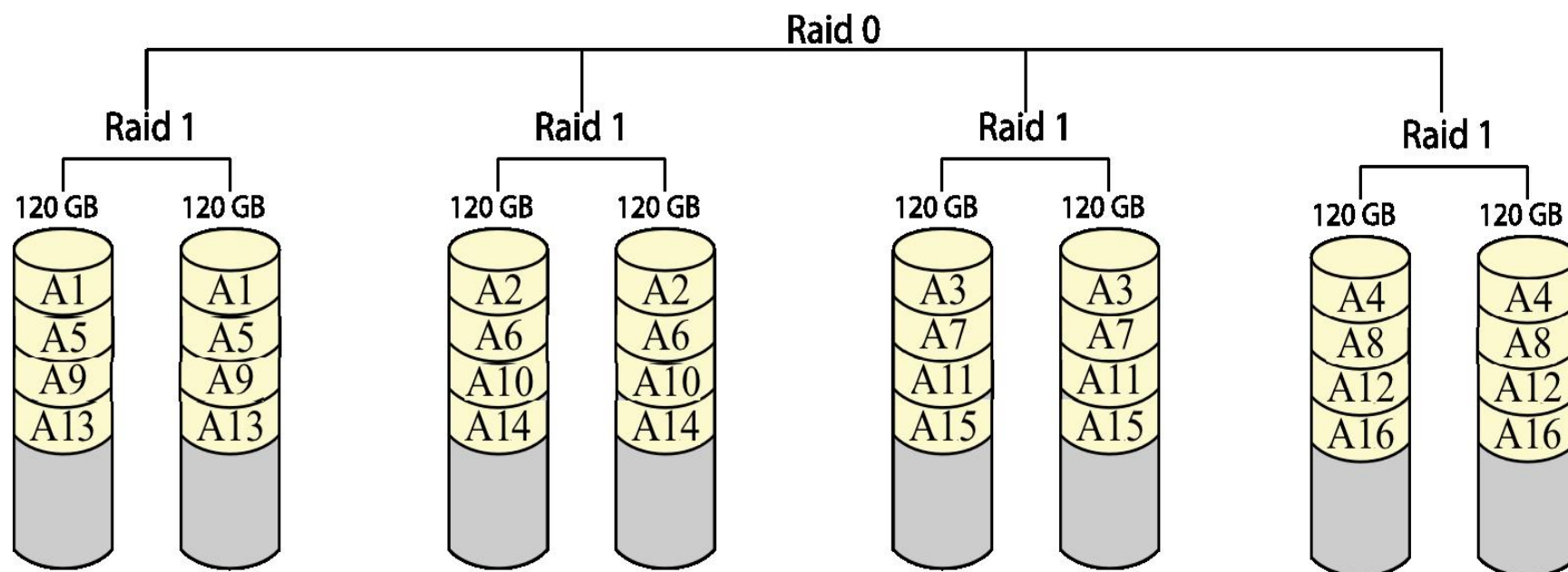
## RAID 1+0 (RAID 10)

Φτιάχνουμε 4 σύνολα των 2 δίσκων, το κάθε σύνολο το οργανώνουμε σε RAID-1 (mirroring) και τα 4 σύνολα σε RAID-0 (striping)

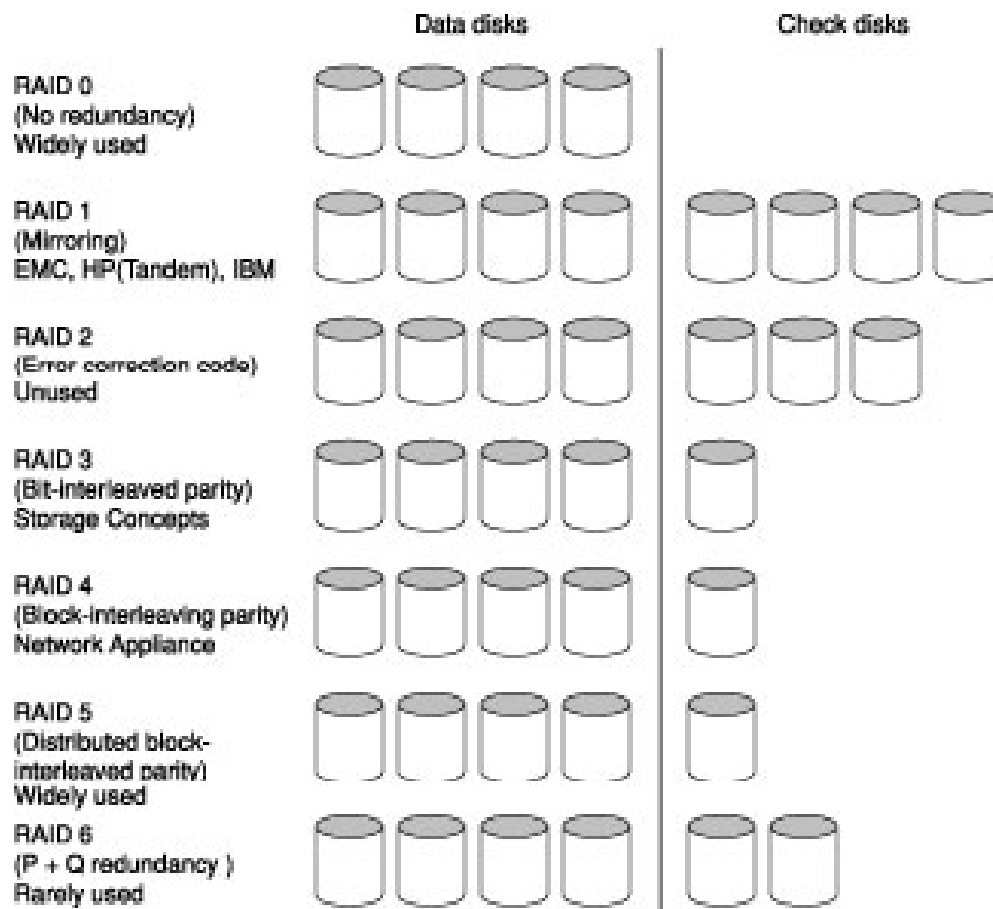
# RAID 0+1 (RAID 01)



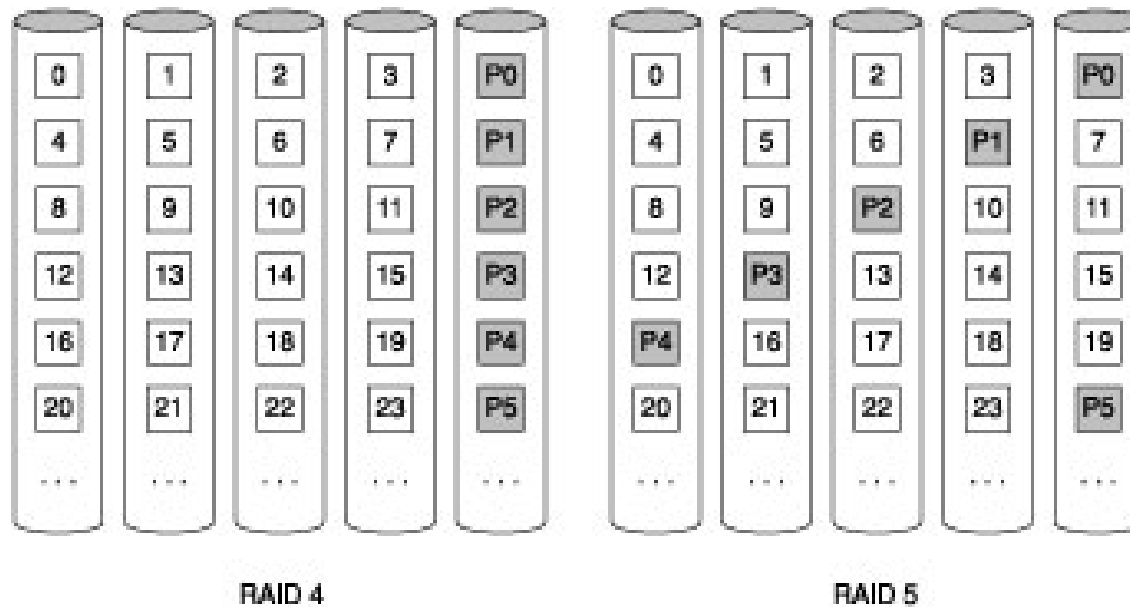
# RAID 1+0 (RAID 10)



# Επίπεδα RAID



# RAID-4 vs RAID-5

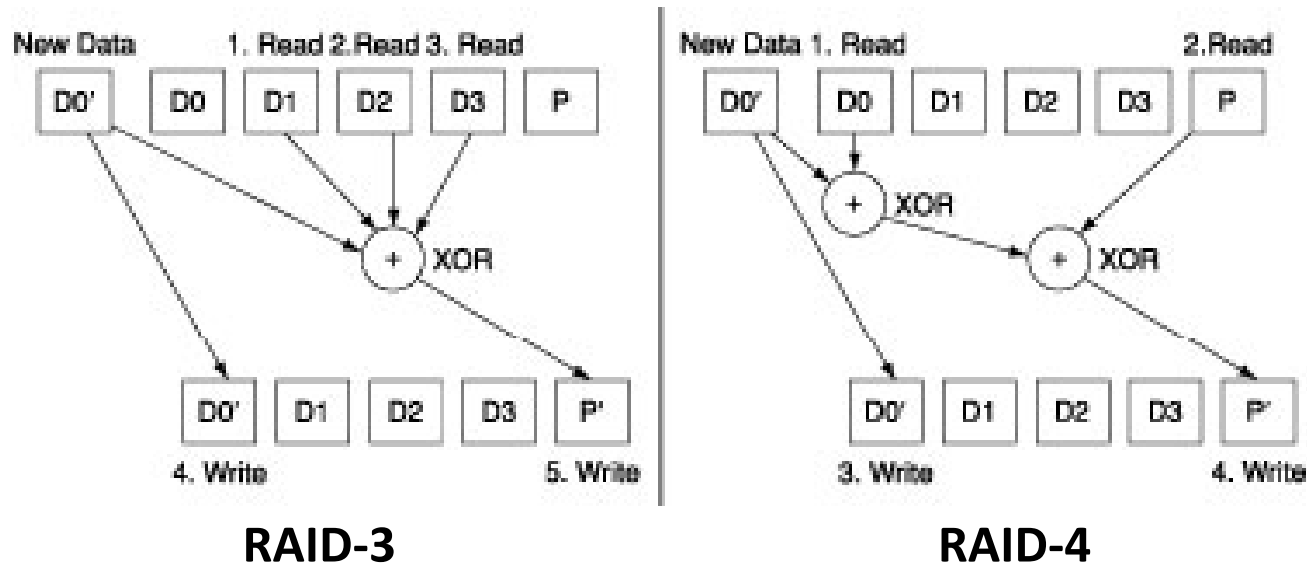


**Block-interleaved parity (RAID-4)**

**Distributed block interleaved parity (RAID-5)**

# Small write update:

Τι γίνεται όταν γράφεται ένα block με νέα τιμή  $D0'$



**RAID-3:** Υπολογισμός νέου Parity  $P'$  αναγκάζει σε ανάγνωση όλους τους δίσκους (3 disk reads ( $D1, D2, D3$ ) και 2 disk writes ( $D0', P'$ ))

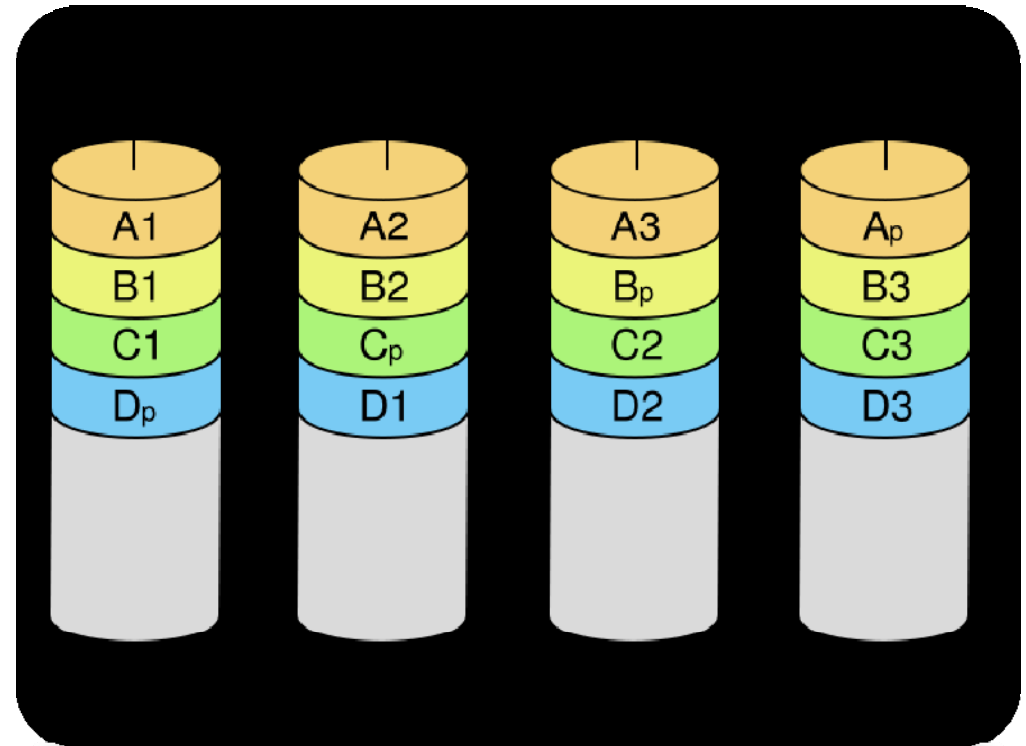
**RAID-4:** Υπολογισμός νέου Parity  $P'$  κάνει ανάγνωση σε 2 δίσκους (2 disk reads ( $D0, P$ ) και 2 disk writes ( $D0', P'$ ))

# RAID 5 - Striped set with distributed parity

Συνεχόμενα blocks γράφονται εναλλάξ στους δίσκους, ενώ κατανέμεται σε αυτούς και ένα block ισοτιμίας.

Παρέχει υψηλή απόδοση στις αναγνώσεις, αφού αυτές μπορούν να γίνουν από πολλούς δίσκους εναλλάξ.

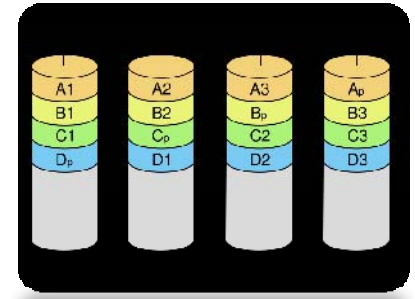
Παρέχει αξιοπιστία, αφού αν πάθει βλάβη ένας δίσκος, τα δεδομένα μπορούν να ανακτηθούν από τους υπόλοιπους .



# RAID 5 - Striped set with distributed parity

**Παράδειγμα:** Έστω ότι διαθέτουμε 4 δίσκους.  
Πώς δουλεύει το RAID 5;

**Απάντηση:** Ας θεωρήσουμε ότι οι 4 δίσκοι έχουν τα  
Παρακάτω δεδομένα (δυαδικό):



	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	0010	
STRIPE1	0010	0000		0100
STRIPE2	0011		1010	1000
STRIPE3		0001	1101	1010

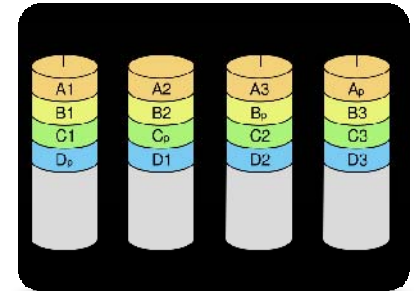
Στα κίτρινα σημεία, τοποθετούνται τα δεδομένα ισοτιμίας. Η ισοτιμία υπολογίζεται ως το **Exclusive-OR (XOR)** του ίδιου stripe όλων των δίσκων.



# RAID 5 - Striped set with distributed parity

**Παράδειγμα:** Έστω ότι διαθέτουμε 4 δίσκους.  
Πώς δουλεύει το RAID 5;

**Απάντηση:** Ας θεωρήσουμε ότι οι 4 δίσκοι έχουν τα Παρακάτω δεδομένα (δυαδικό):



Για όσους δε θυμούνται της XOR ... ο πίνακας αληθείας

XOR		
Είσοδος		Έξοδος
0	0	0
0	1	1
1	0	1
1	1	0

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	0010	
STRIPE1	0010	0000		0100
STRIPE2	0011		1010	1000
STRIPE3		0001	1101	1010

Στα κίτρινα σημεία, τοποθετούνται τα δεδομένα ισοτιμίας. Η ισοτιμία υπολογίζεται ως το **Exclusive-OR (XOR)** του ίδιου stripe όλων των δίσκων.

# RAID 5 - Striped set with distributed parity

---

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	0010	0011
STRIPE1	0010	0000		0100
STRIPE0	0011		1010	1000
STRIPE3		0001	1101	1010

STRIPE0,DISK3 = 0100 XOR 0101 XOR 0010 = 0011

# RAID 5 - Striped set with distributed parity

---

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	0010	0011
<b>STRIPE1</b>	<b>0010</b>	<b>0000</b>	<b>0110</b>	<b>0100</b>
STRIPE2	0011		1010	1000
STRIPE3		0001	1101	1010

STRIPE0,DISK3 = 0100 XOR 0101 XOR 0010 = 0011

STRIPE1,DISK2 = 0010 XOR 0000 XOR 0100 = 0110

# RAID 5 - Striped set with distributed parity

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	0010	0011
STRIPE1	0010	0000	0110	0100
<b>STRIPE2</b>	<b>0011</b>	<b>0001</b>	<b>1010</b>	<b>1000</b>
STRIPE3		0001	1101	1010

STRIPE0,DISK3 = 0100 XOR 0101 XOR 0010 = 0011

STRIPE1,DISK2 = 0010 XOR 0000 XOR 0100 = 0110

STRIPE2,DISK1 = 0011 XOR 1010 XOR 1000 = 0001

# RAID 5 - Striped set with distributed parity

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	0010	0011
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
<b>STRIPE3</b>	<b>0110</b>	<b>0001</b>	<b>1101</b>	<b>1010</b>

STRIPE0,DISK3 = 0100 XOR 0101 XOR 0010 = 0011

STRIPE1,DISK2 = 0010 XOR 0000 XOR 0100 = 0110

STRIPE2,DISK1 = 0011 XOR 1010 XOR 1000 = 0001

STRIPE3,DISK0 = 0001 XOR 1101 XOR 1010 = 0110

# RAID 5 - Striped set with distributed parity

---

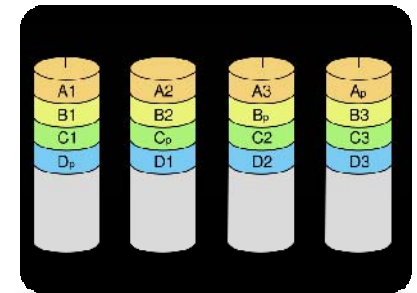
Τελική Εικόνα της συστοιχίας ΔΙΣΚΩΝ  
με διάταξη RAID5

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	0010	0011
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010

# RAID 5 - Striped set with distributed parity

**Παράδειγμα:** Τι γίνεται στις εγγραφές;

**Απάντηση:** Ας θεωρήσουμε ότι οι 4 δίσκοι έχουν τα Παρακάτω δεδομένα (δυαδικό):



	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	0010	0011
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010

Έστω ότι γίνεται η εγγραφή του στοιχείου 1101 στο block 2 (αρίθμηση ξεκινάει από block 0).

# RAID 5 - Striped set with distributed parity

Έστω ότι γίνεται η εγγραφή του στοιχείου 1101 στο block 2 (αρίθμηση ξεκινάει από block 0).

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	0010	0011
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010



# RAID 5 - Striped set with distributed parity

Έστω ότι γίνεται η εγγραφή του στοιχείου 1101 στο block 2 (αρίθμηση ξεκινάει από block 0).

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	<del>0010</del> 1101	0011
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010

Ο ελεγκτής RAID κάνει την εγγραφή του στοιχείου στο αντίστοιχο block ...

# RAID 5 - Striped set with distributed parity

Έστω ότι γίνεται η εγγραφή του στοιχείου 1101 στο block 2 (αρίθμηση ξεκινάει από block 0).

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	<del>0010</del> 1101	0011
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010

Ο ελεγκτής RAID κάνει την εγγραφή του στοιχείου στο αντίστοιχο block ... και ταυτόχρονα ξαναδημιουργεί την ισοτιμία για το συγκεκριμένο stripe, χρησιμοποιώντας παλιά τιμή, νέα τιμή και ισοτιμία

$$\text{STRIPE0,DISK3} = \mathbf{0010} \text{ XOR } \mathbf{1101} \text{ XOR } \mathbf{0011} = \mathbf{1100}$$

# RAID 5 - Striped set with distributed parity

Έστω ότι γίνεται η εγγραφή του στοιχείου 1101 στο block 2 (αρίθμηση ξεκινάει από block 0).

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	<del>0010</del> 1101	<del>0011</del> 1100
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010

Ο ελεγκτής RAID κάνει την εγγραφή του στοιχείου στο αντίστοιχο block ... και ταυτόχρονα ξαναδημιουργεί την ισοτιμία για το συγκεκριμένο stripe, χρησιμοποιώντας παλιά τιμή, νέα τιμή και ισοτιμία

$$\text{STRIPE0,DISK3} = 0010 \text{ XOR } 1101 \text{ XOR } 0011 = \mathbf{1100}$$

# RAID 5 - Striped set with distributed parity

Έστω ότι γίνεται η εγγραφή του στοιχείου 1101 στο block 2 (αρίθμηση ξεκινάει από block 0).

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	<b>1101</b>	<b>1100</b>
STRIPE1	0010	0000	<b>0110</b>	0100
STRIPE2	0011	<b>0001</b>	1010	1000
STRIPE3	<b>0110</b>	0001	1101	1010

Ο ελεγκτής RAID κάνει την εγγραφή του στοιχείου στο αντίστοιχο block ... και ταυτόχρονα ξαναδημιουργεί την ισοτιμία για το συγκεκριμένο stripe, χρησιμοποιώντας παλιά τιμή, νέα τιμή και ισοτιμία

$$\text{STRIPE0,DISK3} = 0010 \text{ XOR } 1101 \text{ XOR } 0011 = 1100$$

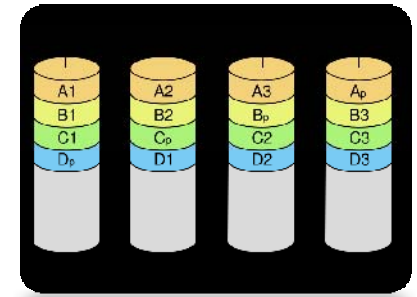
*Η εγγραφή στο RAID 5, ισοδυναμεί με 2 αναγνώσεις και 2 εγγραφές σε δίσκους.*

# RAID 5 - Striped set with distributed parity

**Παράδειγμα:** Τι γίνεται αν χαλάσει ένας δίσκος;

**Απάντηση:** Ας θεωρήσουμε ότι οι 4 δίσκοι έχουν τα Παρακάτω δεδομένα (δυαδικό):

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	1101	1100
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010



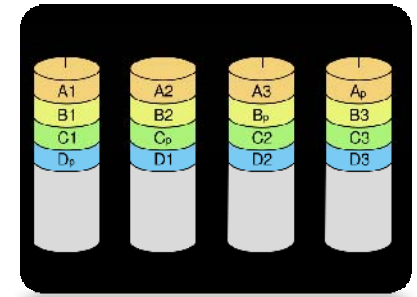
Έστω ότι χαλάει ο DISK2

# RAID 5 - Striped set with distributed parity

**Παράδειγμα:** Τι γίνεται αν χαλάσει ένας δίσκος;

**Απάντηση:** Ας θεωρήσουμε ότι οι 4 δίσκοι έχουν τα Παρακάτω δεδομένα (δυαδικό):

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	1101	1100
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010



Έστω ότι χαλάει ο DISK2

# RAID 5 - Striped set with distributed parity

Έστω ότι χαλάει ο DISK2

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	1101	1100
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010

Ο ελεγκτής RAID εξυπηρετεί τις αιτήσεις για τις πληροφορίες που είχε ο DISK2, χρησιμοποιώντας όλους τους άλλους δίσκους + την ισοτιμία.

# RAID 5 - Striped set with distributed parity

Έστω ότι χαλάει ο DISK2

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	1101	1100
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010

Ο ελεγκτής RAID εξυπηρετεί τις αιτήσεις για τις πληροφορίες που είχε ο DISK2, χρησιμοποιώντας όλους τους άλλους δίσκους + την ισοτιμία.

Έτσι

STRIPE0,DISK2 = 0100 XOR 0101 XOR 1100 = 1101



# RAID 5 - Striped set with distributed parity

Έστω ότι χαλάει ο DISK2

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	1101	1100
STRIPE1	0010	0000	0110	0100
<b>STRIPE2</b>	<b>0011</b>	<b>0001</b>	1010	<b>1000</b>
STRIPE3	0110	0001	1101	1010

Ο ελεγκτής RAID εξυπηρετεί τις αιτήσεις για τις πληροφορίες που είχε ο DISK2, χρησιμοποιώντας όλους τους άλλους δίσκους + την ισοτιμία.

Έτσι

$$\text{STRIPE0,DISK2} = 0100 \text{ XOR } 0101 \text{ XOR } 1100 = 1101$$

$$\text{STRIPE2,DISK2} = 0011 \text{ XOR } 0001 \text{ XOR } 1000 = 1010$$

# RAID 5 - Striped set with distributed parity

Έστω ότι χαλάει ο DISK2

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	1101	1100
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010

Ο ελεγκτής RAID εξυπηρετεί τις αιτήσεις για τις πληροφορίες που είχε ο DISK2, χρησιμοποιώντας όλους τους άλλους δίσκους + την ισοτιμία.

Έτσι

$$\text{STRIPE0,DISK2} = 0100 \text{ XOR } 0101 \text{ XOR } 1100 = 1101$$

$$\text{STRIPE2,DISK2} = 0011 \text{ XOR } 0001 \text{ XOR } 1000 = 1010$$

$$\text{STRIPE3,DISK2} = 0110 \text{ XOR } 0001 \text{ XOR } 1010 = 1101$$

# RAID 5 - Striped set with distributed parity

Έστω ότι χαλάει ο DISK2

	DISK0	DISK1	DISK2	DISK3
STRIPE0	0100	0101	1101	1100
STRIPE1	0010	0000	0110	0100
STRIPE2	0011	0001	1010	1000
STRIPE3	0110	0001	1101	1010

Ο ελεγκτής RAID εξυπηρετεί τις αιτήσεις για τις πληροφορίες που είχε ο DISK2, χρησιμοποιώντας όλους τους άλλους δίσκους + την ισοτιμία.

Έτσι

$$\text{STRIPE0,DISK2} = 0100 \text{ XOR } 0101 \text{ XOR } 1100 = 1101$$

$$\text{STRIPE2,DISK2} = 0011 \text{ XOR } 0001 \text{ XOR } 1000 = 1010$$

$$\text{STRIPE3,DISK2} = 0110 \text{ XOR } 0001 \text{ XOR } 1010 = 1101$$

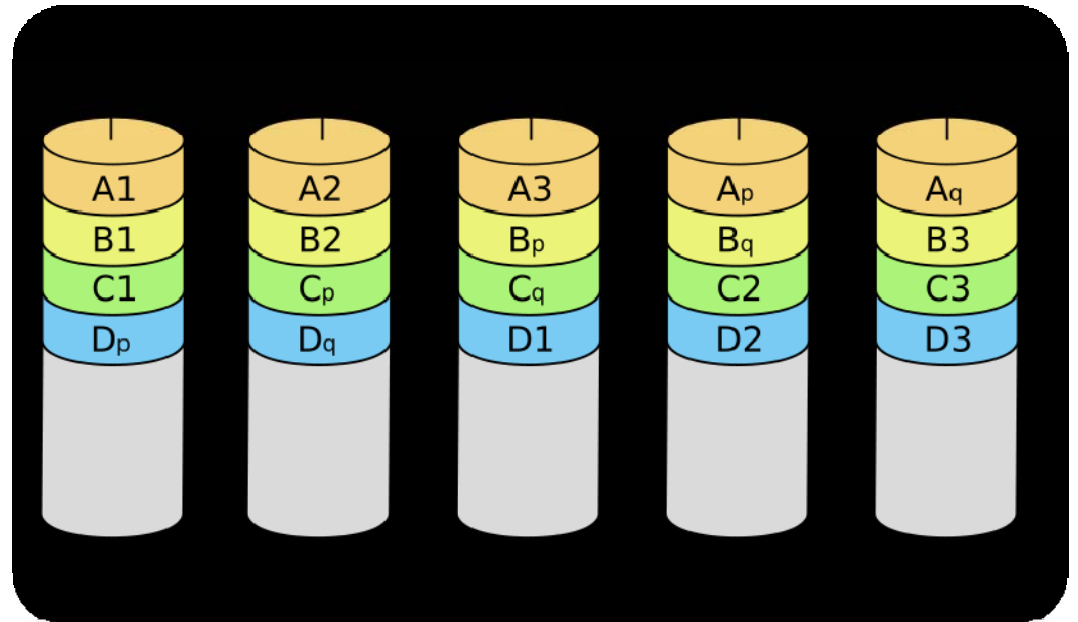
*Κάθε ανάγνωση του χαλασμένου δίσκου, αντιστοιχεί σε αναγνώσεις σε όλους τους υπόλοιπους δίσκους. Καλό είναι να αντικαταστήσουμε το χαλασμένο δίσκο γρήγορα!*

# RAID 6 - Striped set with dual parity

Συνεχόμενα blocks γράφονται εναλλάξ στους δίσκους, ενώ κατανέμεται σε αυτούς και δύο block ισοτιμίας.

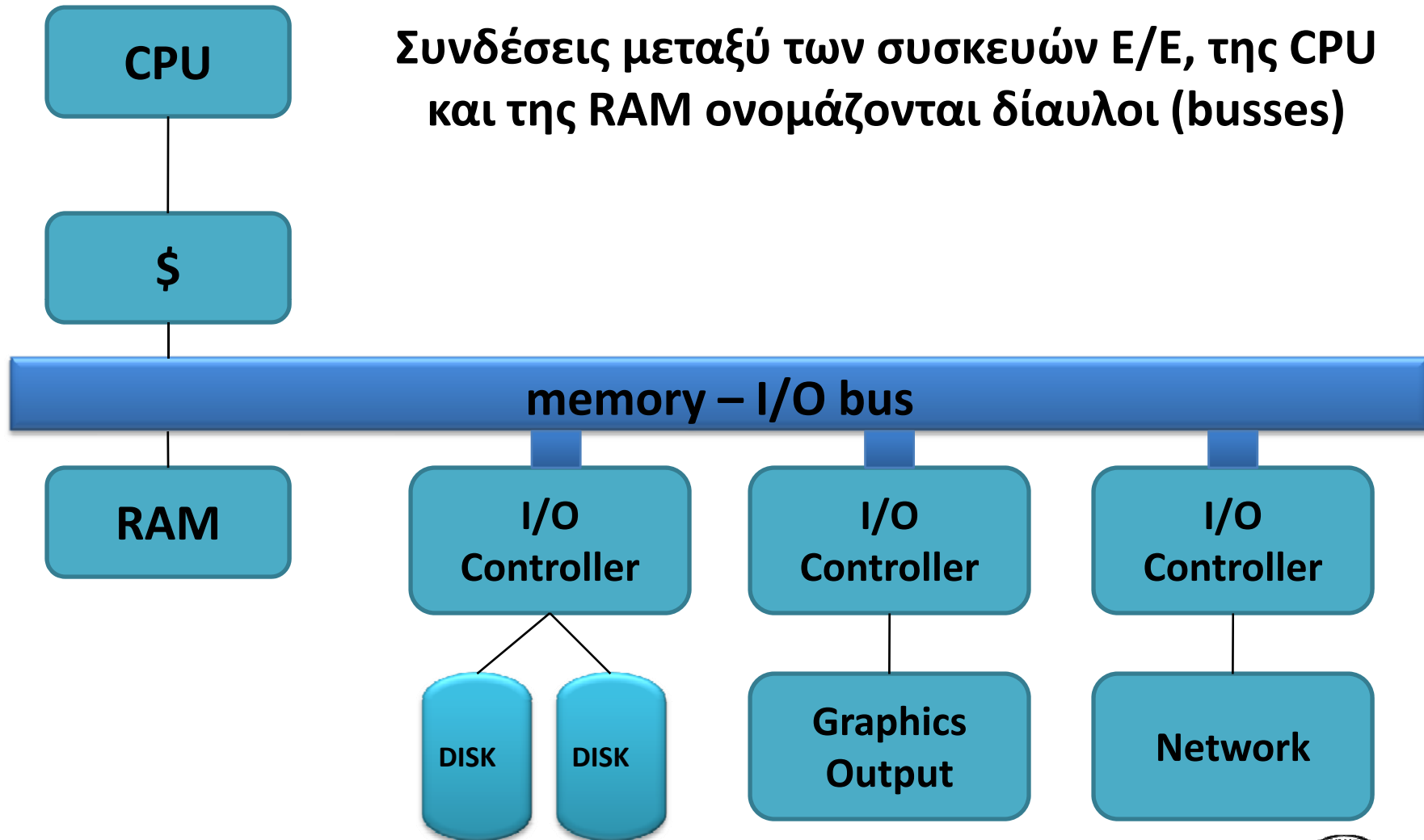
Παρέχει υψηλή απόδοση στις αναγνώσεις, αφού αυτές μπορούν να γίνουν από πολλούς δίσκους εναλλάξ.

Παρέχει αξιοπιστία, αφού αν χαλάσουν μέχρι 2 δίσκοι, τα δεδομένα μπορούν να ανακτηθούν από τους υπόλοιπους .



# Δίαυλοι

Συνδέσεις μεταξύ των συσκευών E/E, της CPU και της RAM ονομάζονται δίαυλοι (busses)

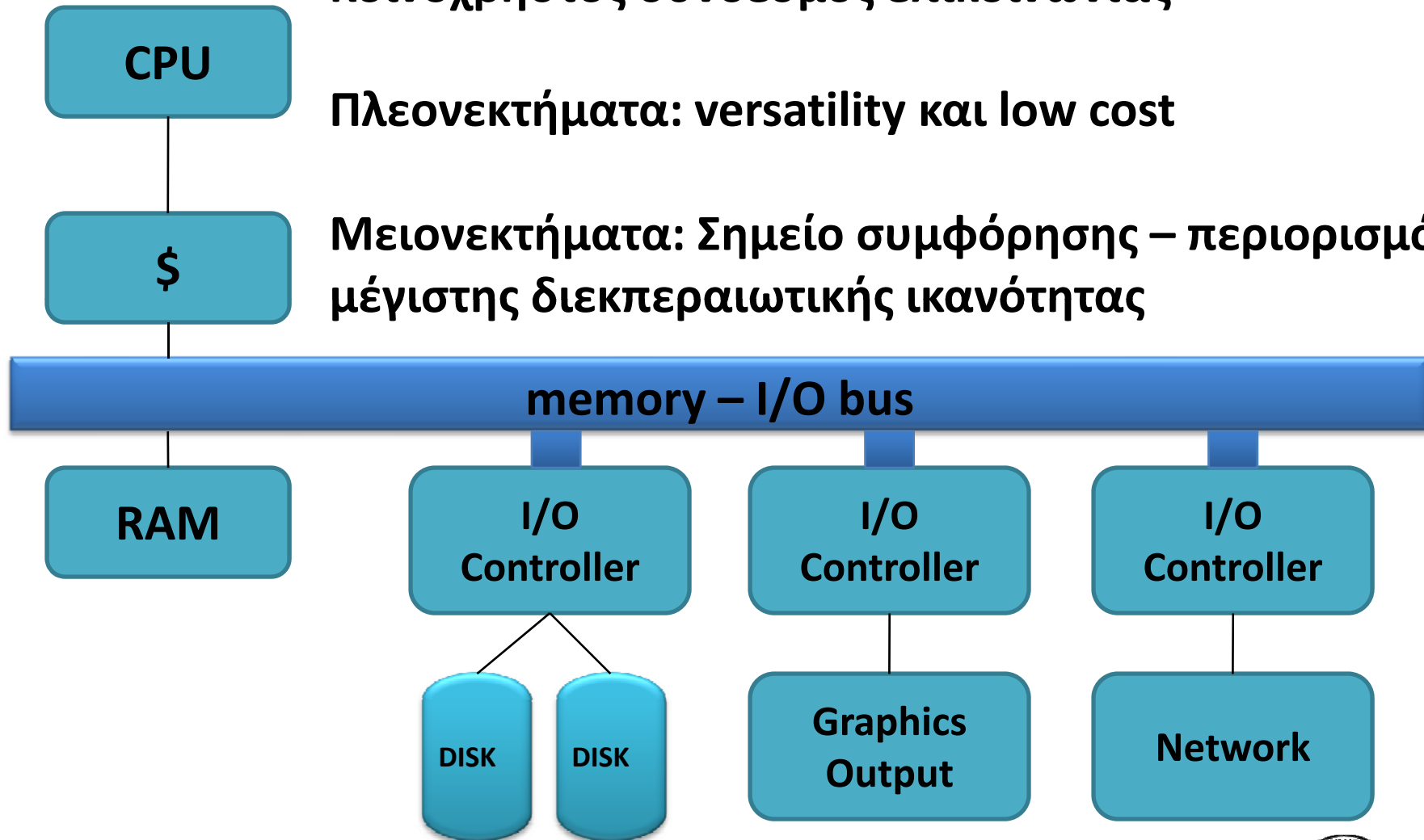


# Δίαυλοι

Κοινόχρηστος σύνδεσμος επικοινωνίας

Πλεονεκτήματα: versatility και low cost

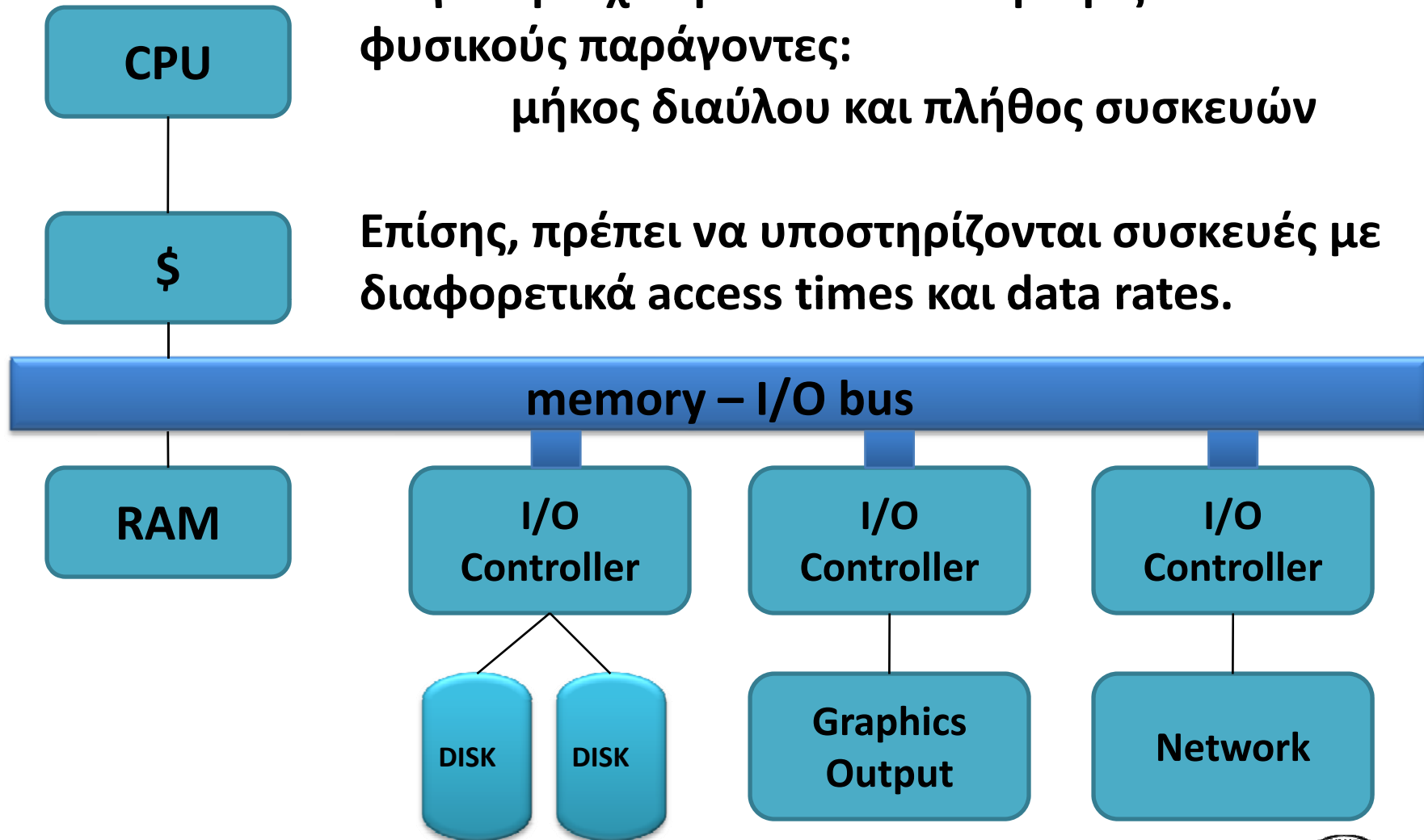
Μειονεκτήματα: Σημείο συμφόρησης – περιορισμός μέγιστης διεκπεραιωτικής ικανότητας



# Δίαυλοι

Μέγιστη ταχύτητα διαύλου περιορίζεται από φυσικούς παράγοντες:  
μήκος διαύλου και πλήθος συσκευών

Επίσης, πρέπει να υποστηρίζονται συσκευές με διαφορετικά access times και data rates.



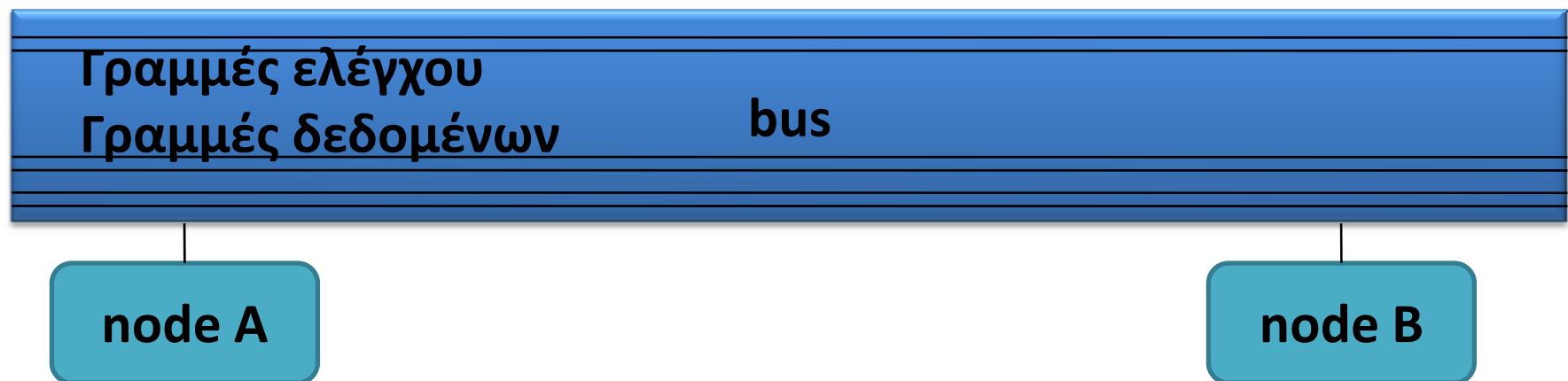
# Βασικά των Διαύλων

---

**Γραμμές ελέγχου:** Σηματοδοτούν requests και acknowledgements

**Γραμμές δεδομένων:** Μεταφέρουν πληροφορίες (δεδομένα ή διευθύνσεις)

(Πολλές φορές υπάρχουν ξεχωριστές γραμμές διευθύνσεων – address vs data bus)





# Βασικά των Διαύλων

---

**Bus transaction:** Ακολουθία λειτουργιών διαύλου που περιλαμβάνει αίτηση (+απόκριση) και μεταφορά δεδομένων

**CPU-memory bus:** Δίαυλος που συνδέει τον επεξεργαστή με τη μνήμη. Μικρό μήκος, υψηλή ταχύτητα, μεγιστοποίηση CPU-memory bandwidth

**I/O bus:** Μεγάλο μήκος, πολλές συνδεδεμένες συσκευές, ποικιλία στο εύρος ζώνης δεδομένων συσκευών

# Βασικά των Διαύλων

---

Οι συσκευές E/E δεν συνδέονται απευθείας στη μνήμη. Δεδομένα πάνε μέσω του διαύλου CPU-memory (backplane bus).

Λόγω αυξημένων απαιτήσεων, έχουν δημιουργηθεί ειδικοί δίαυλοι (π.χ. γραφικών)

I/O bus χρησιμεύει για την σύνδεση νέων περιφερειακών. Ανάπτυξη προτύπων (π.χ. USB, Firewire) για διασφάλιση λειτουργίας συσκευών σε όλα τα συστήματα.

# Βασικά χαρακτηριστικά USB/Firewire

Χαρακτηριστικά	Firewire (1394)	USB 2.0
Τύπος διαύλου	Εισόδου/Εξόδου	Εισόδου/Εξόδου
Βασικό εύρος διαύλου δεδομένων (σήματα)	4	2
Χρονισμός	Ασύγχρονος	Ασύγχρονος
Μέγιστο θεωρητικό bandwidth	50MB/sec (Firewire 400) ή 100MB/sec (Firewire 800)	0,2MB/sec (χαμηλή ταχύτητα) ή 1,5MB/sec (πλήρης ταχύτητα) ή 60MB/sec (υψηλή ταχύτητα)
Hotplug	NAI	NAI
Μέγιστος αριθμός συσκευών	63	127
Μέγιστο μήκος διαύλου (χάλκινο καλώδιο)	4,5μ	5μ
Όνομα προτύπου	IEEE 1394, 1394b	USB Implementers Forum

# Μέθοδοι επικοινωνίας διαύλων

---

## Σύγχρονη vs Ασύγχρονη Επικοινωνία

**Σύγχρονη επικοινωνία: Ρολόι στις γραμμές ελέγχου**

**Μειονεκτήματα: Όλες οι συσκευές πρέπει να έχουν το ίδιο ρολόι – δύσκολο (βλ. clock skew) – εφικτό για μικρούς διαύλους (π.χ. CPU-MEM)**

**Πλεονεκτήματα: Υψηλή Ταχύτητα**

# Μέθοδοι επικοινωνίας διαύλων

---

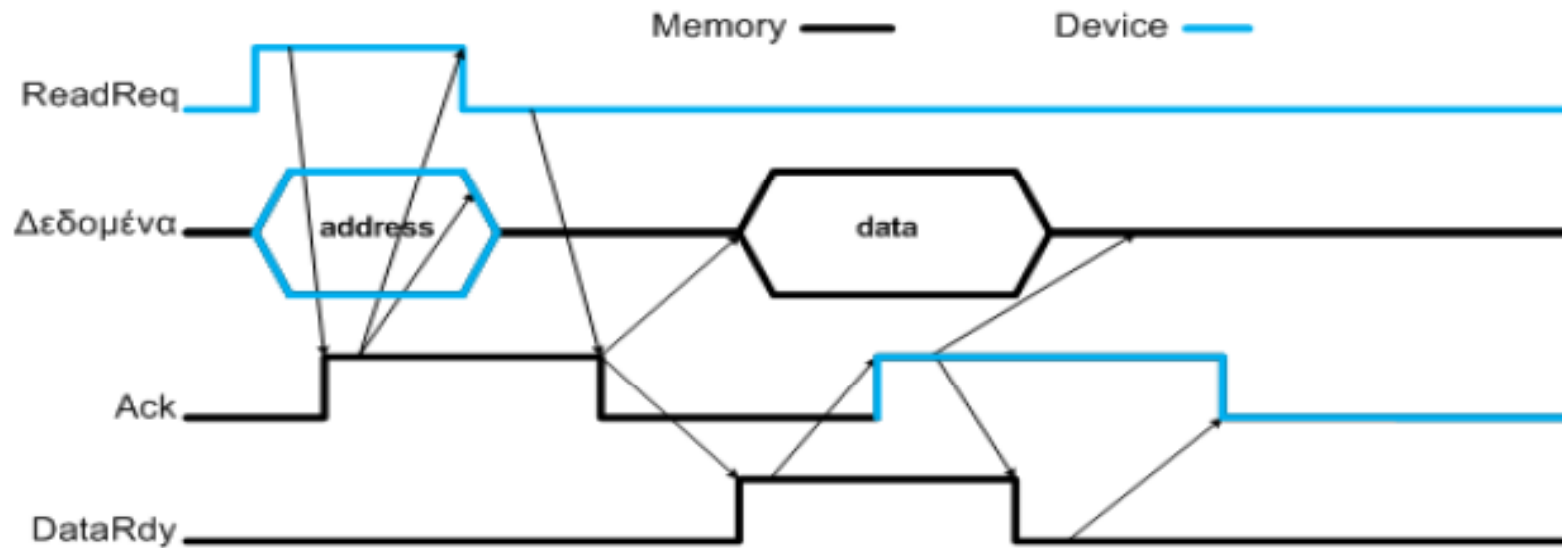
## Σύγχρονη vs Ασύγχρονη Επικοινωνία

**Ασύγχρονη επικοινωνία: Χωρίς ρολόι**

**Πλεονεκτήματα: Μεγάλη ποικιλία συσκευών, μεγάλο μήκος (USB, Firewire = asynchronous)**

**Πλεονεκτήματα: Χαμηλή Ταχύτητα (σε σχέση με CPU-MEM bus)**

# Παράδειγμα ασύγχρονης επικοινωνίας

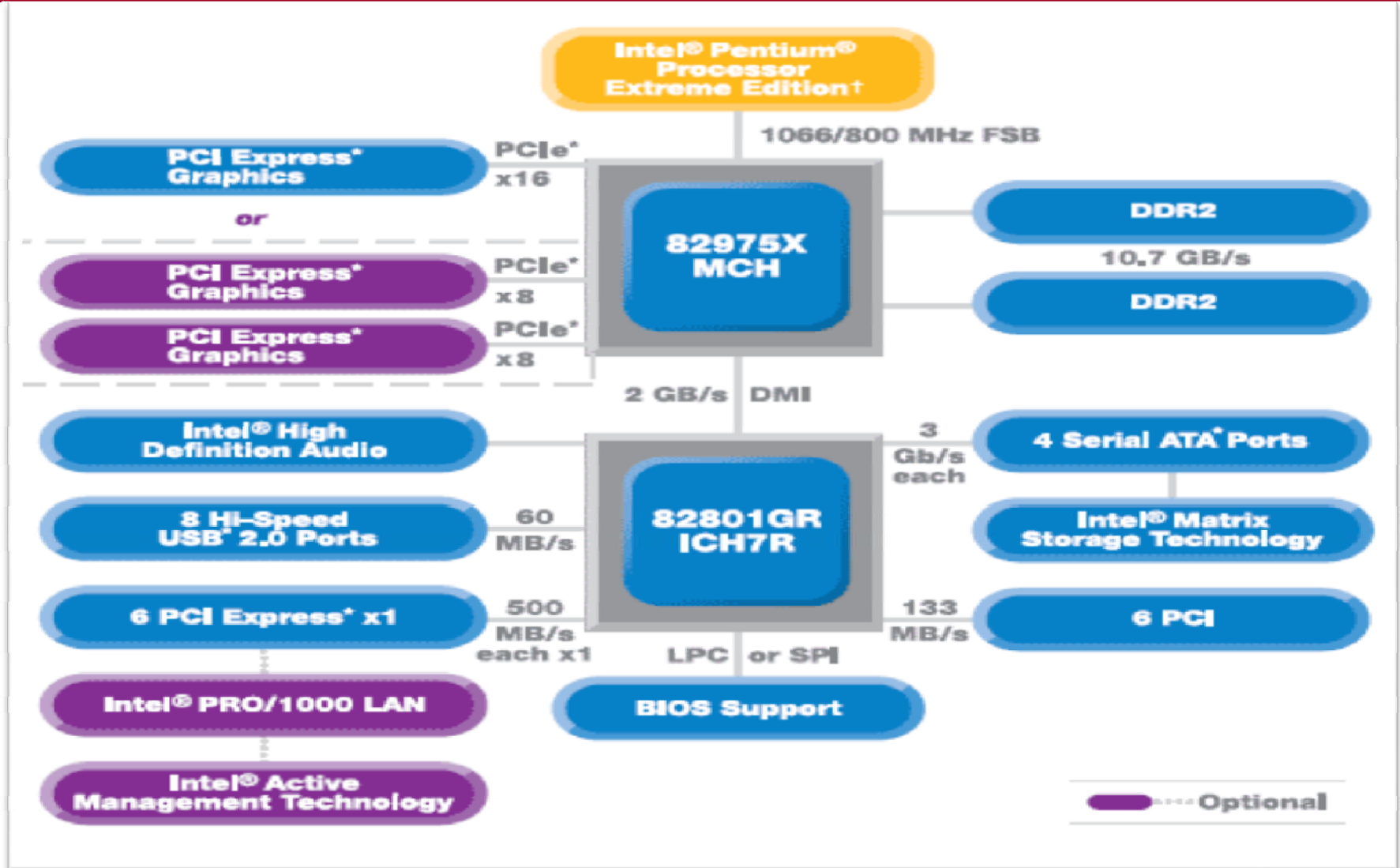


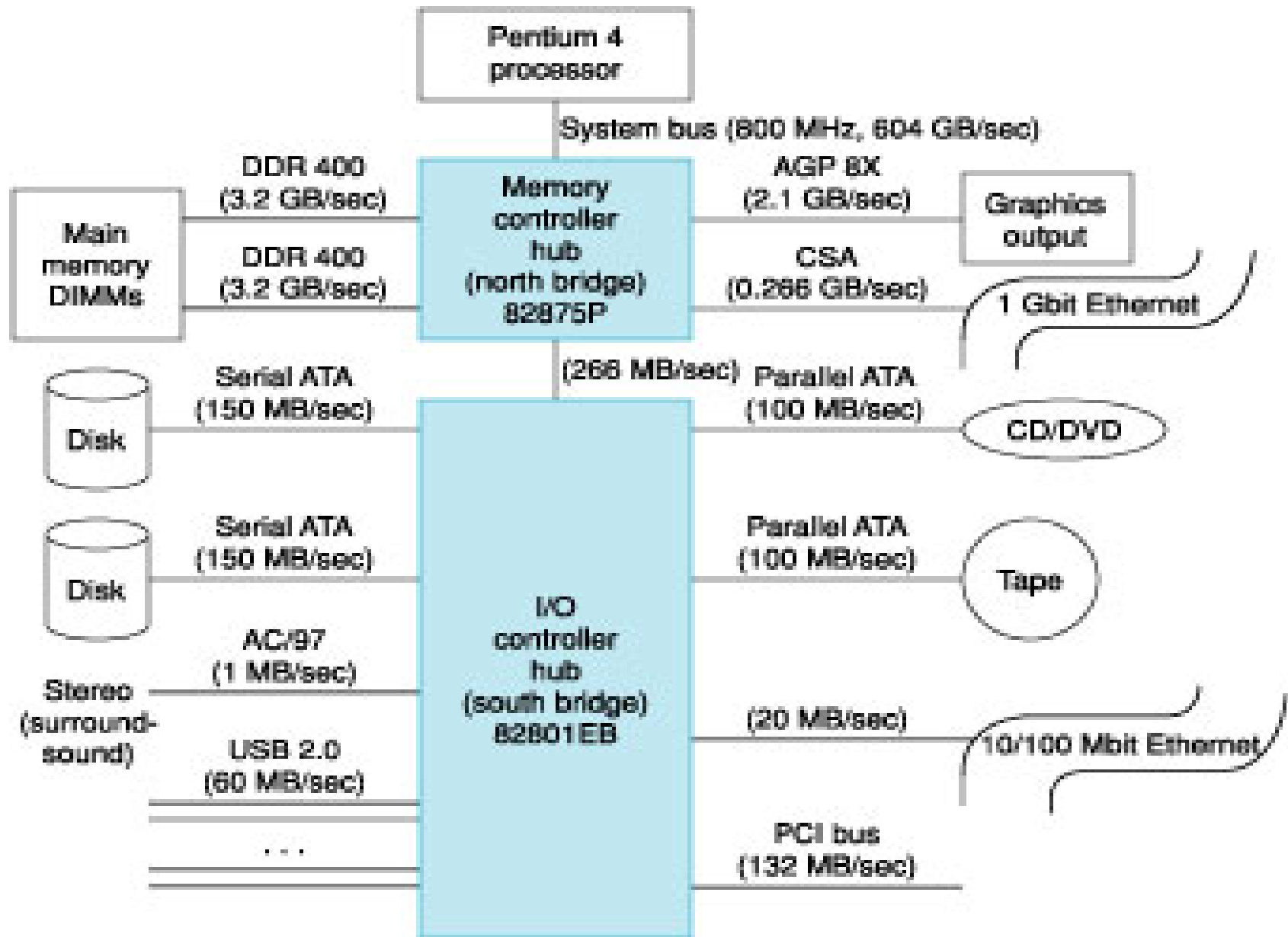
**ReadReq:** Δηλώνει αίτηση ανάγνωσης. Διεύθυνση τοποθετείται στη γραμμή δεδομένων.

**DataRdy:** Δηλώνει έγκυρα δεδομένα στη γραμμή δεδομένων

**Ack:** Επιβεβαιώνει το σήμα ReadReq ή DataRdy της άλλης πλευράς

# Core2Duo busses and interconnects







# Διαταγές προς συσκευές Ε/Ε

---

Για να μιλήσει ο επεξεργαστής με I/O χρησιμοποιούνται 2 μέθοδοι:

## α) memory mapped I/O

π.χ. Οι διευθύνσεις 0xFFFF0000 – 0xFFFF000F αντιστοιχούν σε 4 command registers 32bit μιας συσκευής I/O. Οι εγγραφές σε αυτές τις διευθύνσεις, αντί για τη μνήμη, γράφουν στους registers.

## β) special Input/Output commands

# Επικοινωνία με τον επεξεργαστή

---

Όταν μια συσκευή E/E ολοκληρώσει μια λειτουργία, πώς ενημερώνει τη CPU;

**α) Polling** – η CPU ελέγχει ανά τακτά χρονικά διαστήματα τη συσκευή για αλλαγές στο state

**β) Interrupt** – η συσκευή EE στέλνει ένα interrupt (διακοπή) στη CPU για να «τραβήξει» την προσοχή της

# Μεταφορά δεδομένων μεταξύ ΕΕ και μνήμης

---

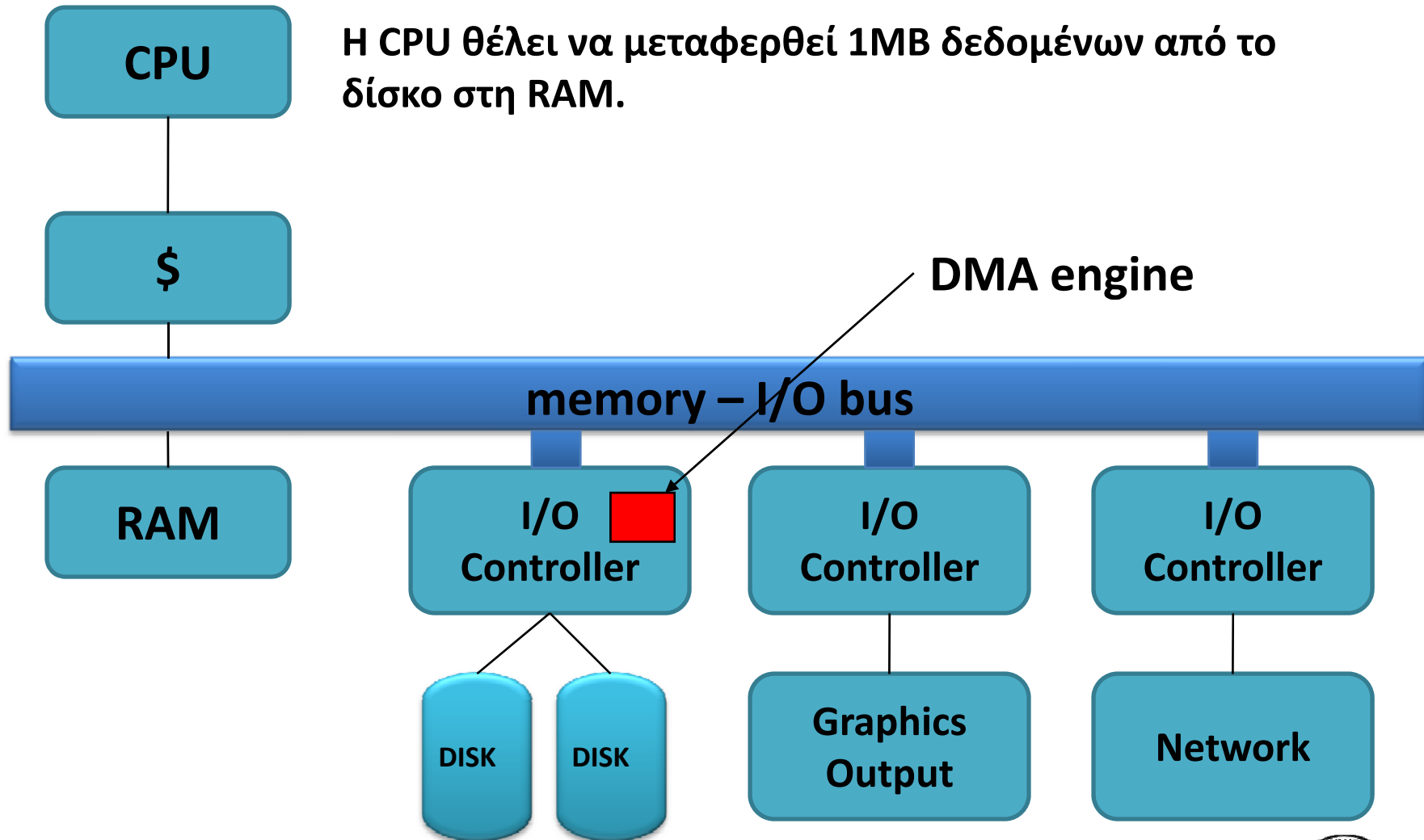
Πώς μεταφέρονται τα δεδομένα από μια συσκευή (π.χ. δίσκο) στη μνήμη του υπολογιστή;

**α) Programmed I/O** – η CPU διαβάζει τα δεδομένα και τα γράφει στη μνήμη (**CPU busy**)

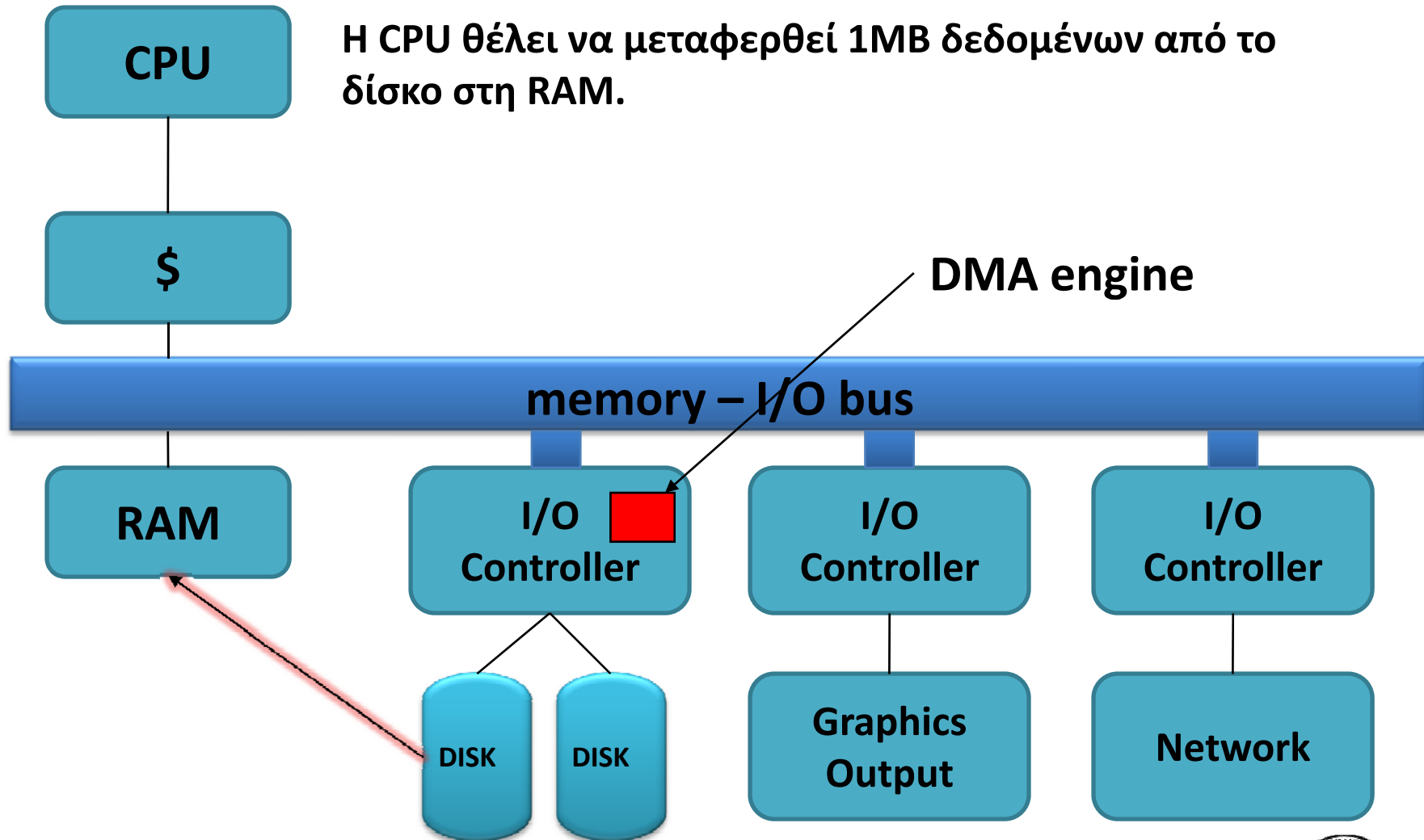
**β) Direct Memory Access** – η CPU προγραμματίζει τη συσκευή DMA μιας συσκευής να γράψει μόνη της τα δεδομένα στη μνήμη (bus master) και όταν ολοκληρώσει να ενημερώσει τη CPU με interrupt (**CPU free**) - προβλήματα

**Η κάθε μονάδα DMA λειτουργεί σαν μια μικρή ειδική CPU για το σκοπό της μεταφοράς δεδομένων απευθείας στη μνήμη χωρίς τη μεσολάβηση του επεξεργαστή**

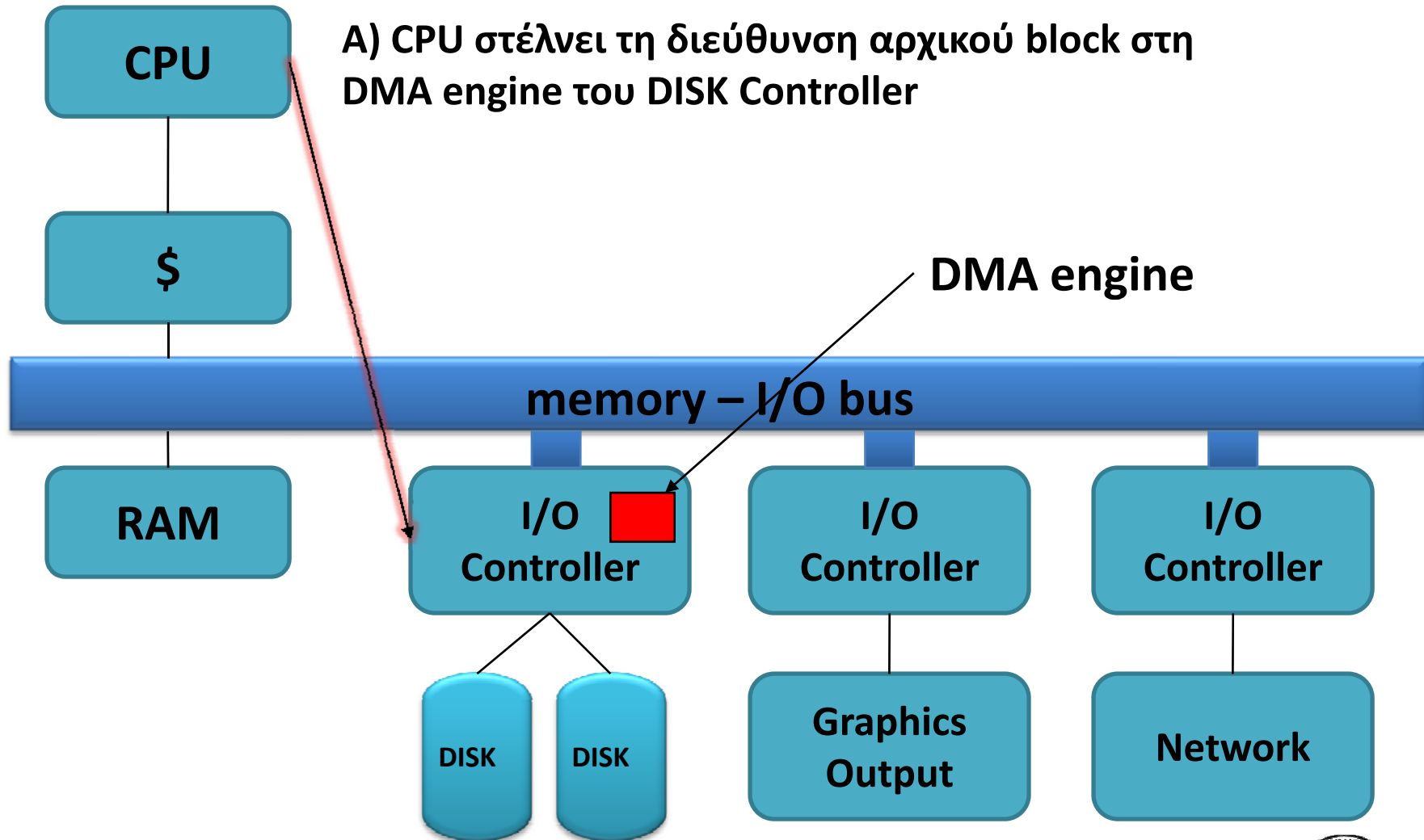
# Direct Memory Access



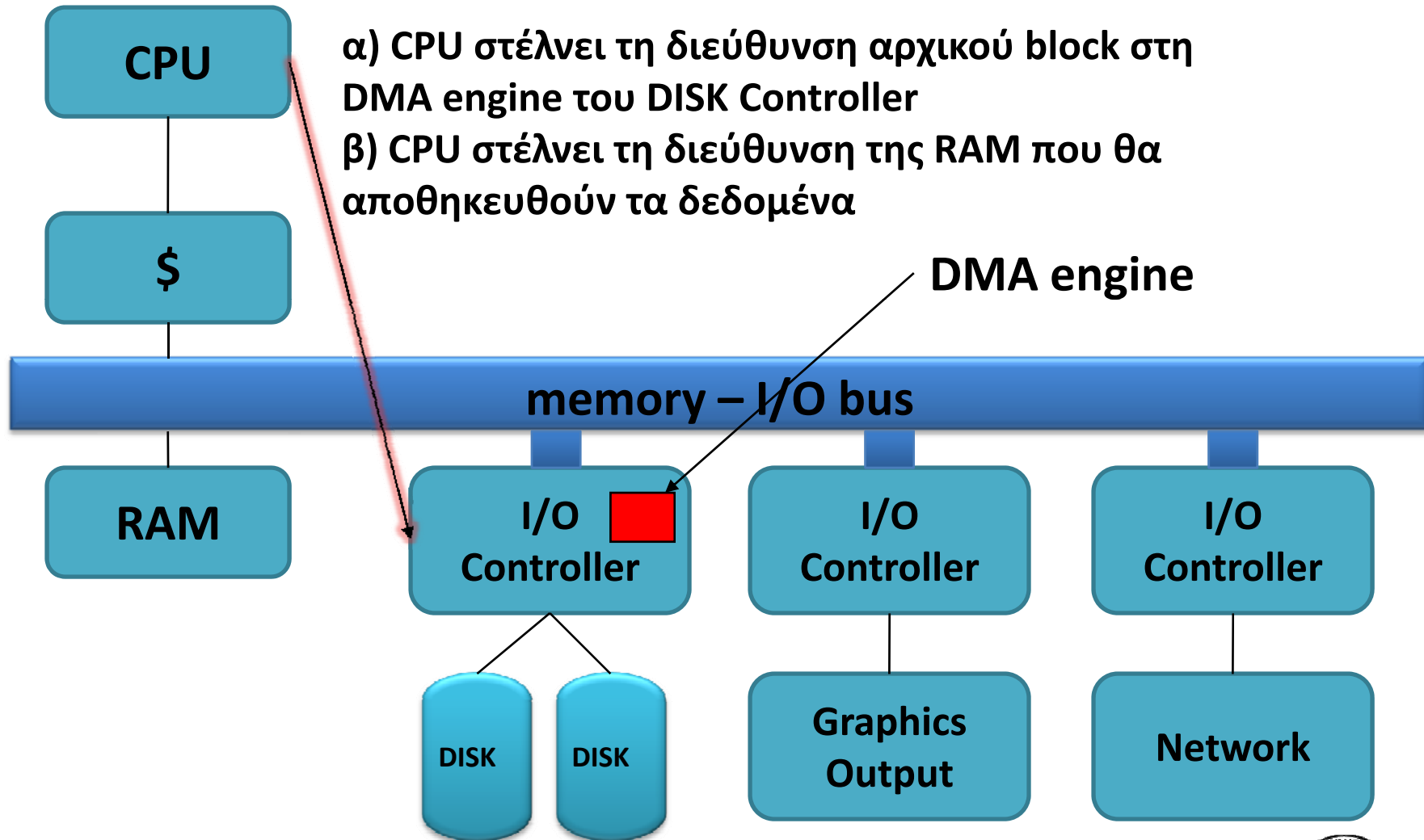
# Direct Memory Access



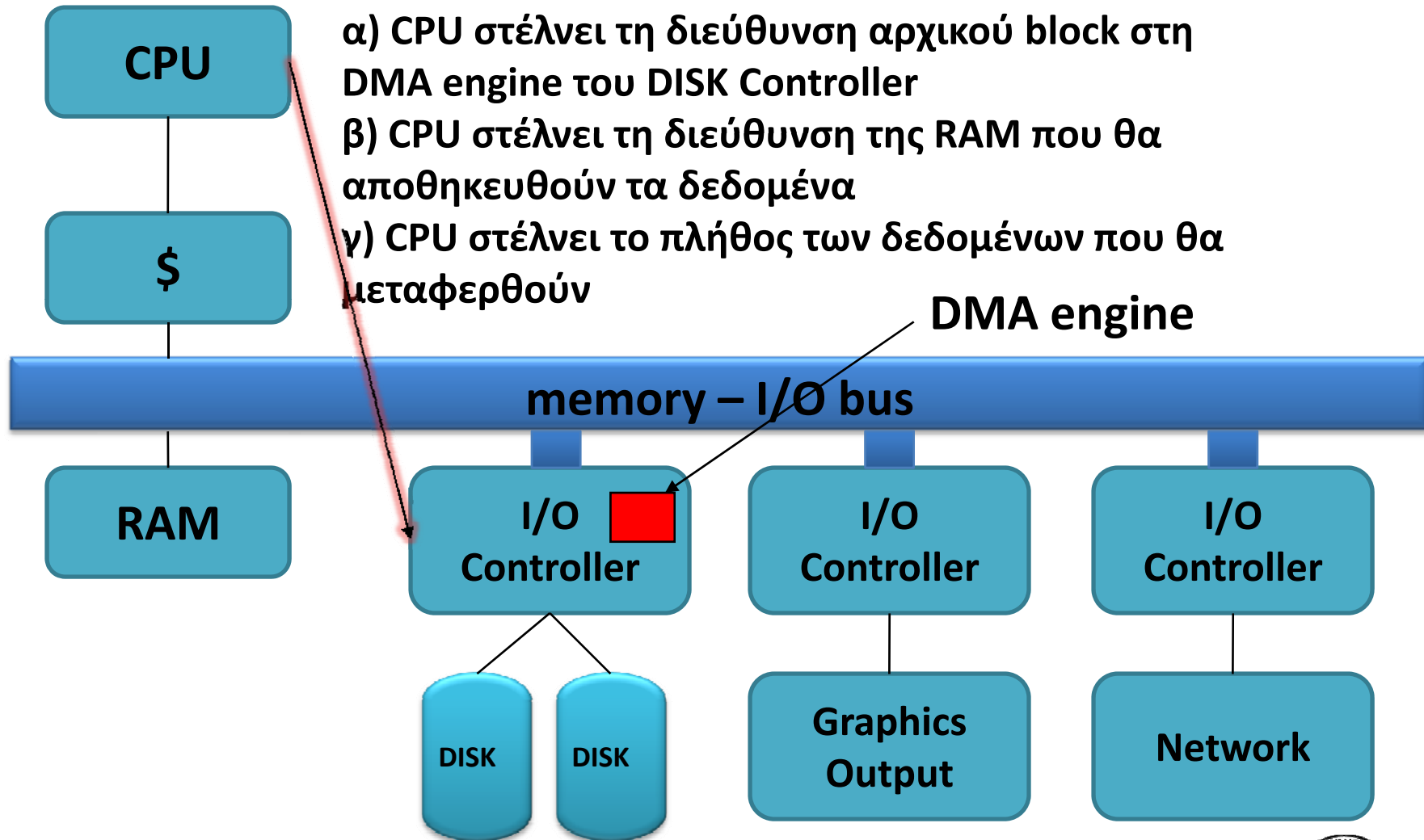
# Direct Memory Access



# Direct Memory Access

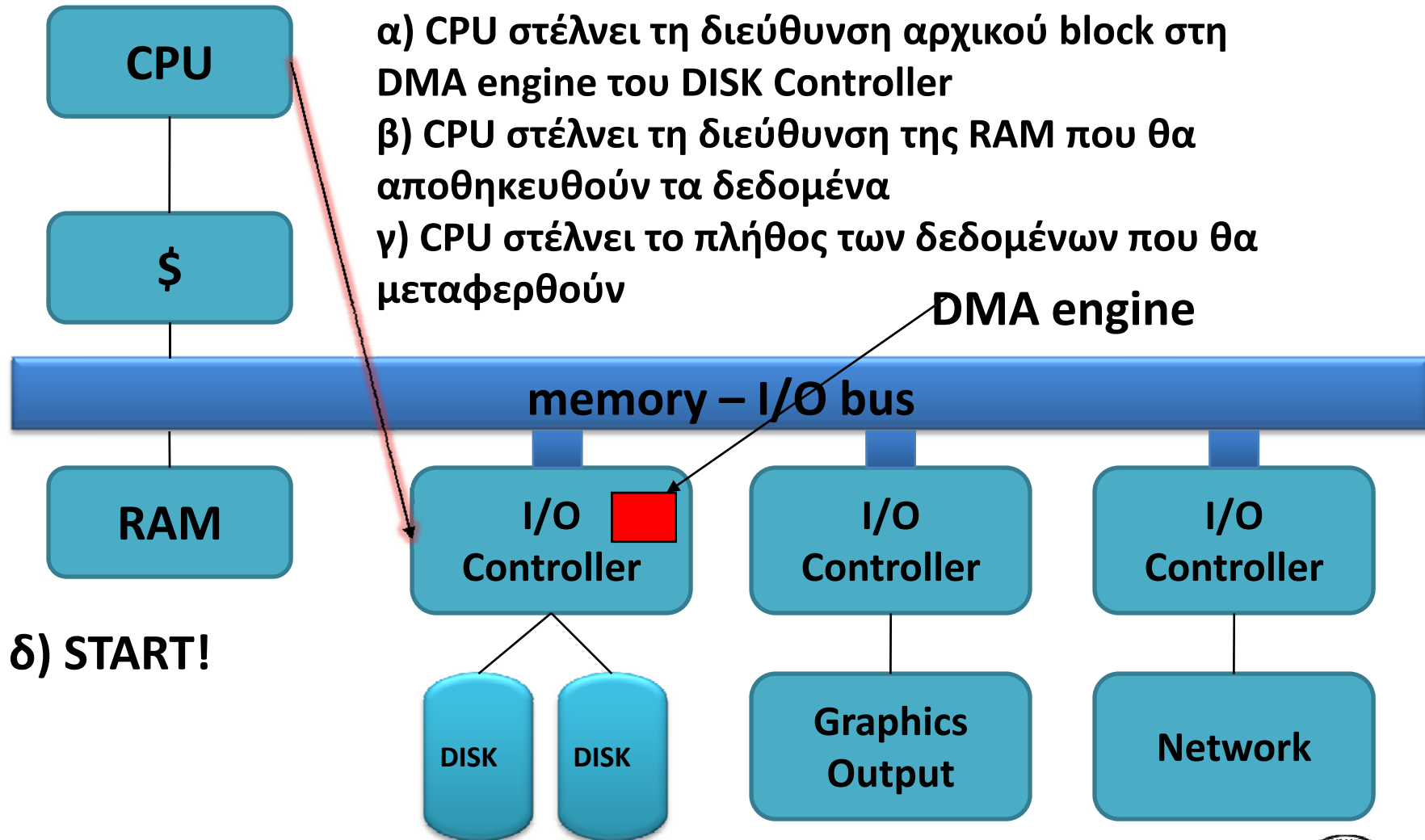


# Direct Memory Access

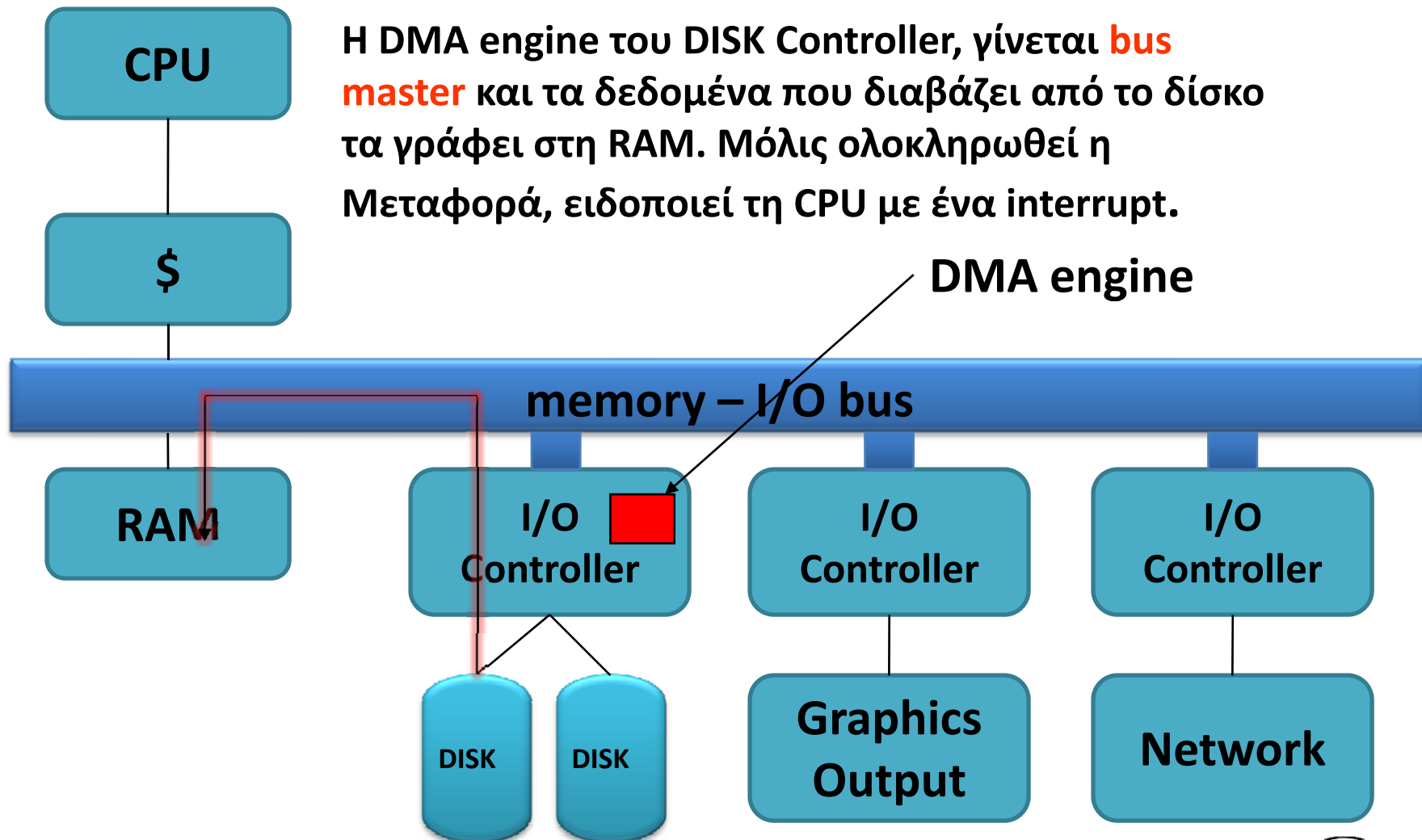




# Direct Memory Access



# Direct Memory Access



Η DMA engine του DISK Controller, γίνεται **bus master** και τα δεδομένα που διαβάζει από το δίσκο τα γράφει στη RAM. Μόλις ολοκληρωθεί η Μεταφορά, ειδοποιεί τη CPU με ένα interrupt.

DMA engine

memory - I/O bus

RAM

I/O Controller

DISK

DISK

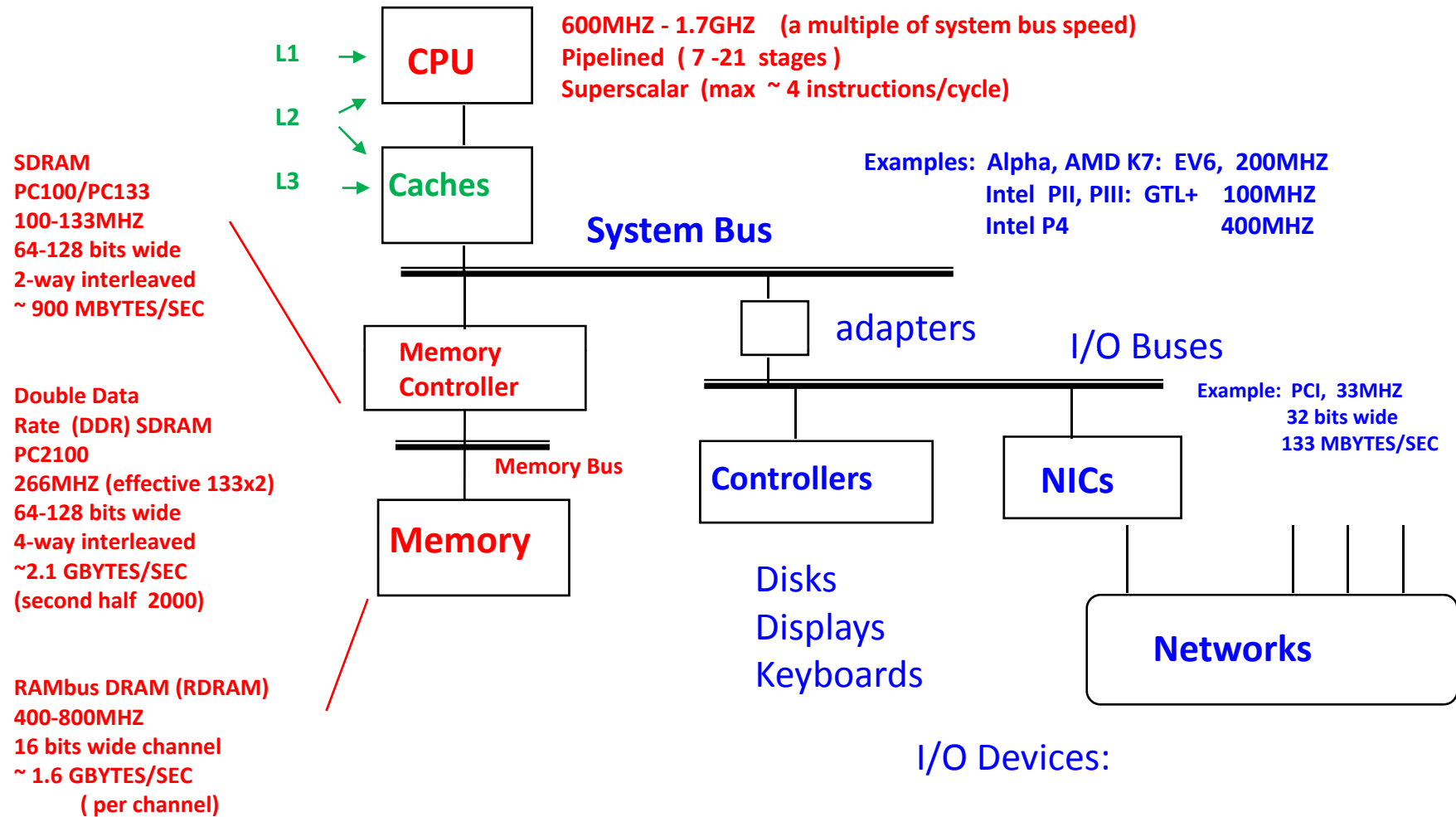
I/O Controller

Graphics Output

I/O Controller

Network

# Συστατικά ενός Computer System



# Intel Hub Architecture (850 Chipset)

---

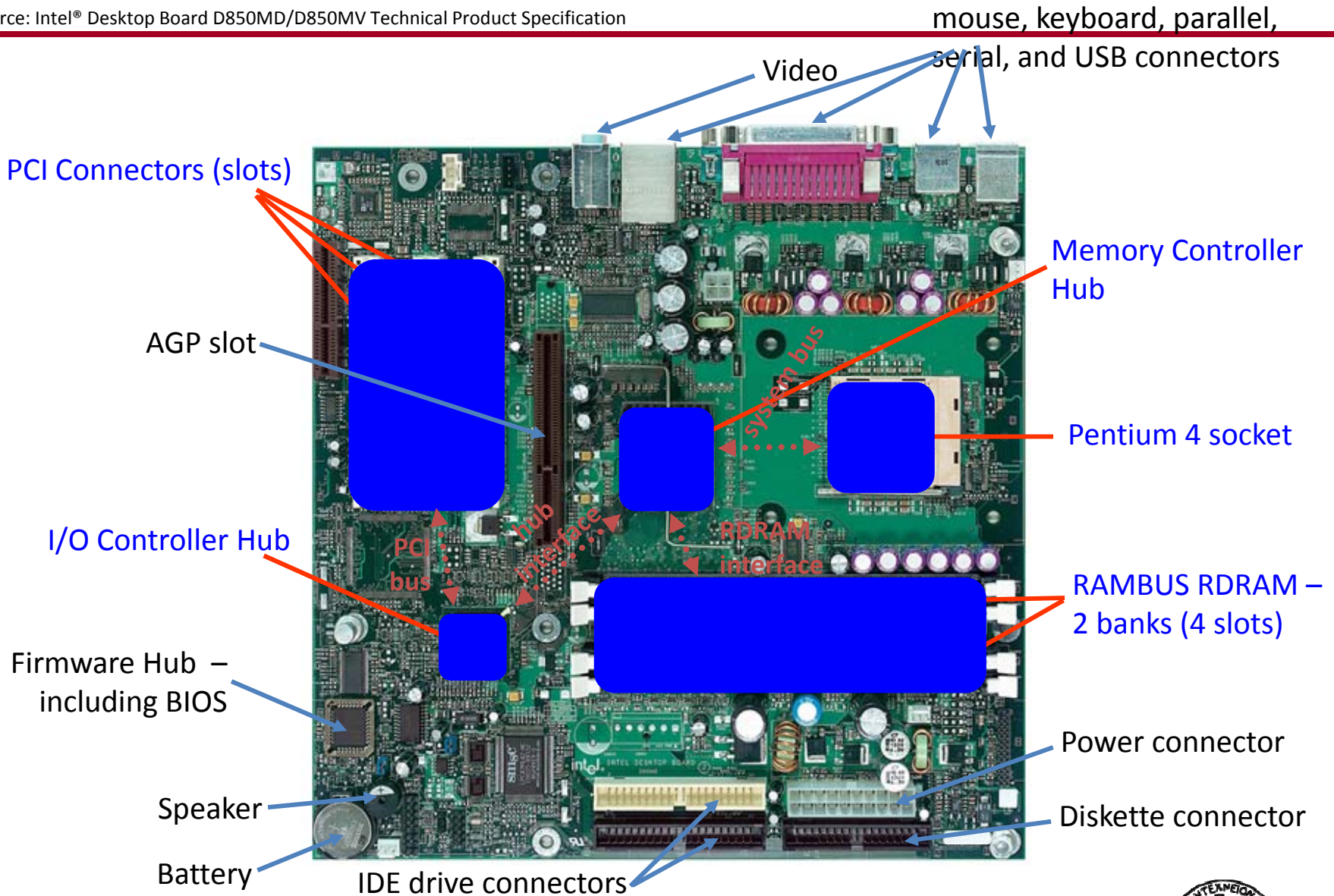


## Intel D850MD Motherboard:

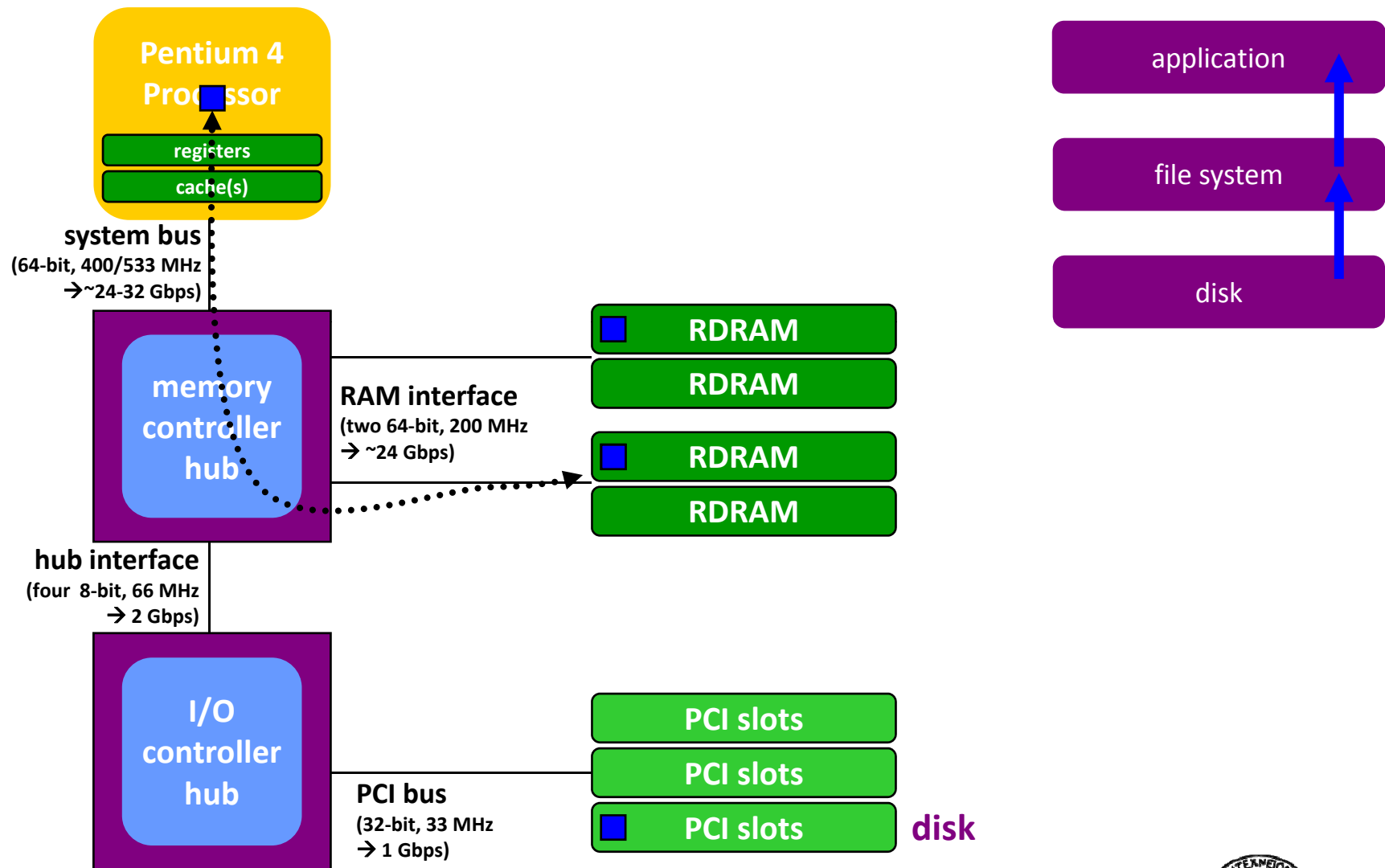
Source: Intel® Desktop Board D850MD/D850MV Technical Product Specification

# Intel D850MD Motherboard:

Source: Intel® Desktop Board D850MD/D850MV Technical Product Specification



# Example: Intel Hub Architecture (850 Chipset)



# Intel 32-bit Architecture (IA32): Basic Execution Environment

- **Address space:**  $1 - 2^{36}$  (64 GB),  
 each program may have a linear address space of 4 GB ( $2^{32}$ )

## Basic program execution registers:

- 8 general purpose registers (EAX, EBX, ECX, EDX, ESI, EDI, EBP and ESP)
- 6 segment registers (CS, DS, SS, ES, FS and GS)
- 1 flag register (EFLAGS)
- 1 instruction pointer register (EIP)

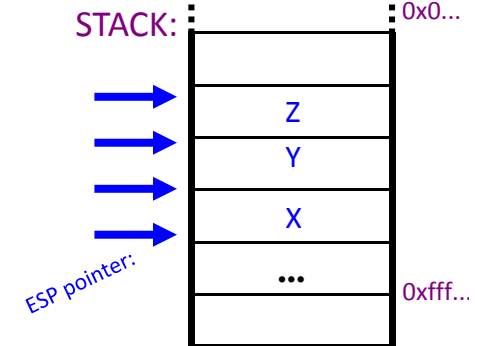
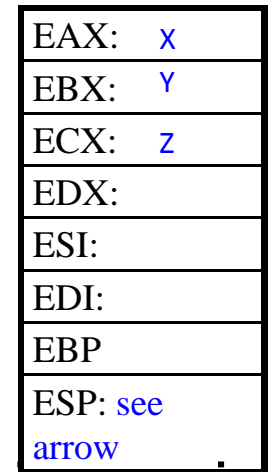
**Stack** – a continuous array of memory locations

- Current stack is referenced by the SS register
- ESP register – stack pointer
- EBP register – stack frame base pointer (fixed reference)
- PUSH – stack grow, add item (ESP decrement)
- POP – remove item, stack shrinks (ESP increment)

Several other registers like Control, MMX, XMM, FPU, MTRR, MSR and performance monitoring

GPRs:

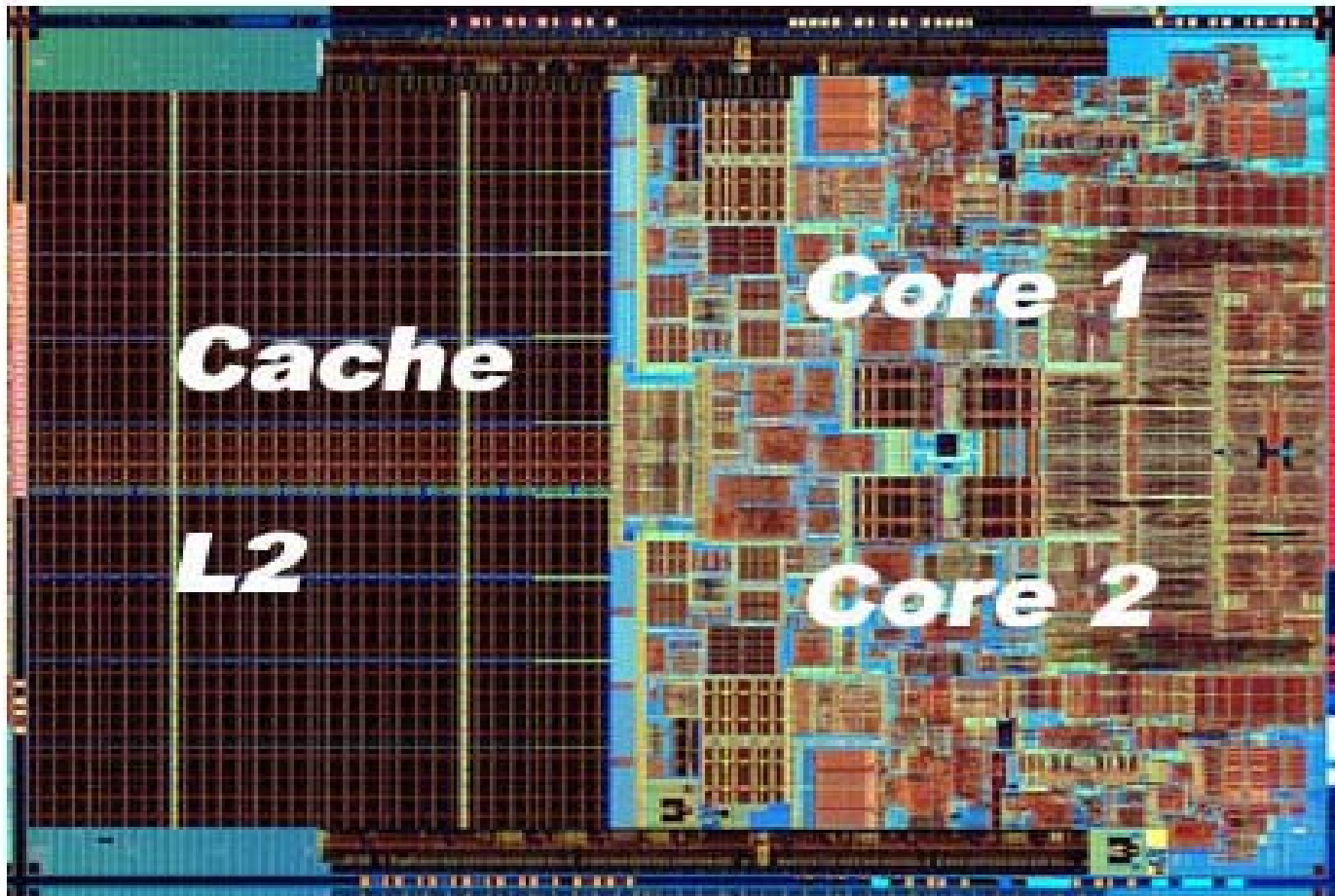
```
PUSH %eax
PUSH %ebx
PUSH %ecx
<do something>
POP %ecx
POP %ebx
POP %eax
```





# Core 2 CPU die (Conroe march)

---





# Intel i975X chipset

