



ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ
ΟΡΓΑΝΩΣΗ ΥΠΟΛΟΓΙΣΤΩΝ (206ΕΥΥΚ)
ΠΠΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΕΑΡΙΝΟ 2023-2024

Διάλεξη Νο4:
M68000: Σύστημα Μνήμης
Δ. Καραμπατζάκης, Επίκουρος Καθηγητής
email. dkara@cs.ihu.gr

Δήλωση προσβασιμότητας

Σε αυτό το μάθημα όλες/οι οι φοιτήτριες/τές απολαμβάνουν – και αντίστοιχα υποχρεούνται να σέβονται – το δικαίωμα της ίσης μεταχείρισης. Δεν είναι ανεκτή και αποδεκτή κανενός τύπου και μορφής διάκριση με κριτήρια την εθνικότητα, τη φυλή, την καταγωγή, τη γλώσσα, το φύλο, τη θρησκεία, την ηλικία, την υγεία, τη σωματική ικανότητα, την ιδιωτική ζωή, τον γενετήσιο προσανατολισμό, τη σωματική ικανότητα και την οικονομική και κοινωνική κατάσταση στην οποία αυτοί βρίσκονται.

Το Πανεπιστήμιο άγρυπνα μεριμνά για τη διασφάλιση της αρχής των ίσων ευκαιριών και της ίσης μεταχείρισης. Οι κοινωνικές προκαταλήψεις και οι ιδεολογικές παρωπίδες είναι έννοιες τελείως ξένες με την επιστημονική πρόοδο την οποία το Πανεπιστήμιο είναι ταγμένο να υπηρετεί.

Ο Διδάσκων

Πληροφορίες για το Μάθημα

Διδάσκων:

Δημήτρης Καραμπατζάκης, Επίκουρος Καθηγητής
Αναλογικά και Ψηφιακά Ηλεκτρονικά Συστήματα
Μέλος Εργαστηρίου Βιομηχανικών και Εκπαιδευτικών
Ενσωματωμένων Συστημάτων

Επικοινωνία / πληροφορίες:

Email. dkara@cs.ihu.gr

web. <http://www.internetofthings.gr/>

Ώρες Γραφείου:

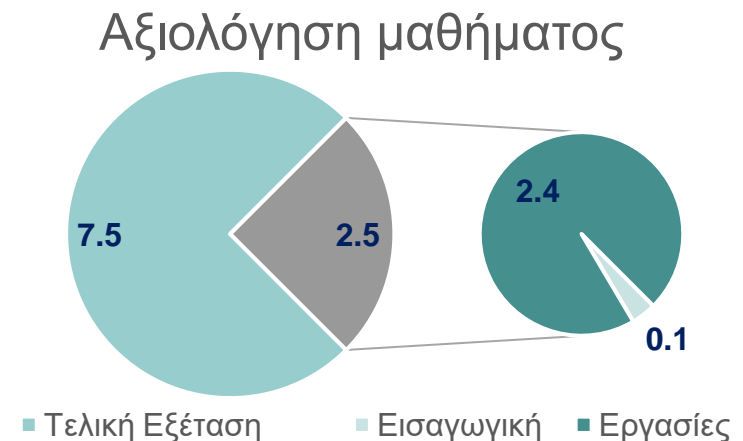
μετά από συνεννόηση με email στο ΦΕ 315 (πάνω από αιθ. Α1)

Πληροφορίες για το Μάθημα (Γενικές)

- Κάθε Τρίτη, Πέμπτη **12.00 π.μ. - 14.00 μ.μ.** μάθημα θεωρίας στο Μεγάλο Αμφιθέατρο (μπορεί να αλλάζει με ανακοινώσεις).
- Η διαχείριση του μαθήματος θα γίνει με χρήση της υπηρεσίας <https://courses.cs.ihu.gr>
- Όλοι οι φοιτητές πρέπει να έχουν λογαριασμό στο [uregister](#).
- Η ιστοσελίδα με τις πληροφορίες του μαθήματος: http://iees.cs.ihu.gr/?page_id=3209
- Υλικό του μαθήματος στο moodle: <https://moodle.cs.ihu.gr/>

Πληροφορίες για το Μάθημα (Αξιολόγηση)

- Η βαθμολογία είναι **75%** από την τελική εξέταση και **25%** από τις ατομικές εργασίες (1 σετ ασκήσεων) που θα δοθούν για το σπίτι.
- Η τελική εξέταση είναι με ανοιχτό το κύριο σύγγραμμα του μαθήματος.
- Ο βαθμός του μαθήματος ($BM = ΓΕ*0,75 + ΣΑ*0,25$) πρέπει να είναι τουλάχιστον πέντε (5).



Πληροφορίες για το Μάθημα (Μονάδες)

- Κωδικός Μαθήματος: 206ΕΥΥΚ
- Εξάμηνο: 2ο
- Τύπος Μαθήματος: Υποβάθρου, Ανάπτυξης Δεξιοτήτων
- Είδος Μαθήματος: Υποχρεωτικό (ΥΠ)
- Διδασκαλία Θεωρίας: 3 ώρες/εβδομάδα
- Διδασκαλία Φροντιστήριο: 1 ώρες/εβδομάδα
- Πιστωτικές μονάδες ECTS: 7
- Γλώσσα διδασκαλίας και Εξετάσεων: Ελληνικά

Πληροφορίες για το Μάθημα (Φόρτος)

● Δραστηριότητα	Φόρτος εργασίας εξαμήνου
● Διαλέξεις	78 ώρες
● Φροντιστηριακές Ασκήσεις	26 ώρες
● Γραπτές Εξετάσεις	2 ώρες
● Γραπτές Εργασίες	34 ώρες
● Αυτοτελής Μελέτη	35 ώρες
● Σύνολο	175 ώρες (7 ECTS)

Κύριο Σύγγραμμα Μαθήματος (ΕΥΔΟΞΟΣ)



Οργάνωση και Σχεδίαση Υπολογιστών

Συγγραφέας: Πογαρίδης Δημήτριος

Έτος Έκδοσης: 2019

Κωδικός στον Εύδοξο: **86192986**

Λογισμικό - Αναπτυξιακό

- **A' μέρος μαθήματος (CISC):**
 - Assembly για τον Motorola68000
 - Λογισμικό easy68k <http://www.easy68k.com/>
- **B' μέρος μαθήματος (RISC):**
 - Υλοποίηση σχεδιάσεων σε αναπτυξιακό Arduino (προαιρετική αγορά του υλικού σύμφωνα με τις οδηγίες)
 - Λογισμικό Arduino IDE
<https://www.arduino.cc/en/Main/Software>
 - Η γλώσσα προγραμματισμού (C++) και οι εντολές που υποστηρίζει είναι διαθέσιμες στο:
<https://www.arduino.cc/reference/en/>

Σύστημα Μνήμης Memory System

Μνήμη

Κάθε σύστημα μνήμης χρειάζεται έναν αριθμό γραμμών εισόδου και εξόδου για να εκτελέσει τις παρακάτω λειτουργίες:

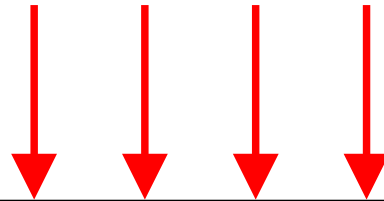
- **Να επιλέξει τη διεύθυνση στη μνήμη, όπου θα γίνει εγγραφή ή από όπου θα γίνει ανάγνωση.**
- **Να επιλέξει αν θα γίνει εγγραφή ή ανάγνωση.**
- **Να τροφοδοτήσει τα δεδομένα που πρόκειται να εναποθηκευτούν στη μνήμη, στην περίπτωση εγγραφής στη μνήμη.**

• Να κρατήσει τα δεδομένα στην έξοδο κατά τη διάρκεια μιας ανάγνωσης.

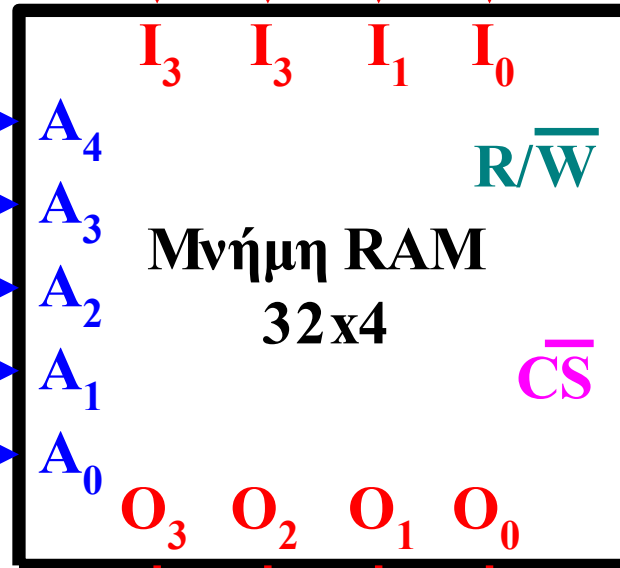
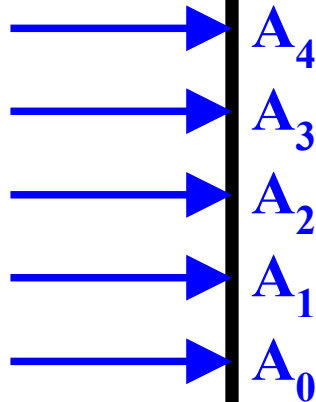
• Να απελευθερώσει ή να δεσμεύσει τη μνήμη στο να ανταποκριθεί ή όχι στις εισόδους διεύθυνσης και στην εντολή ανάγνωση/εγγραφή (Read/Write).

Διάγραμμα μνήμης

Είσοδοι Δεδομένων
(Data Inputs)



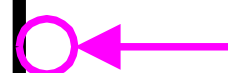
Είσοδοι
Διευθύνσεων
(Address
Inputs)



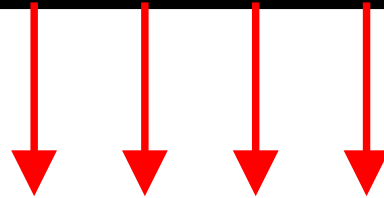
Ανάγνωση/Εγγραφή
(Read/Write)



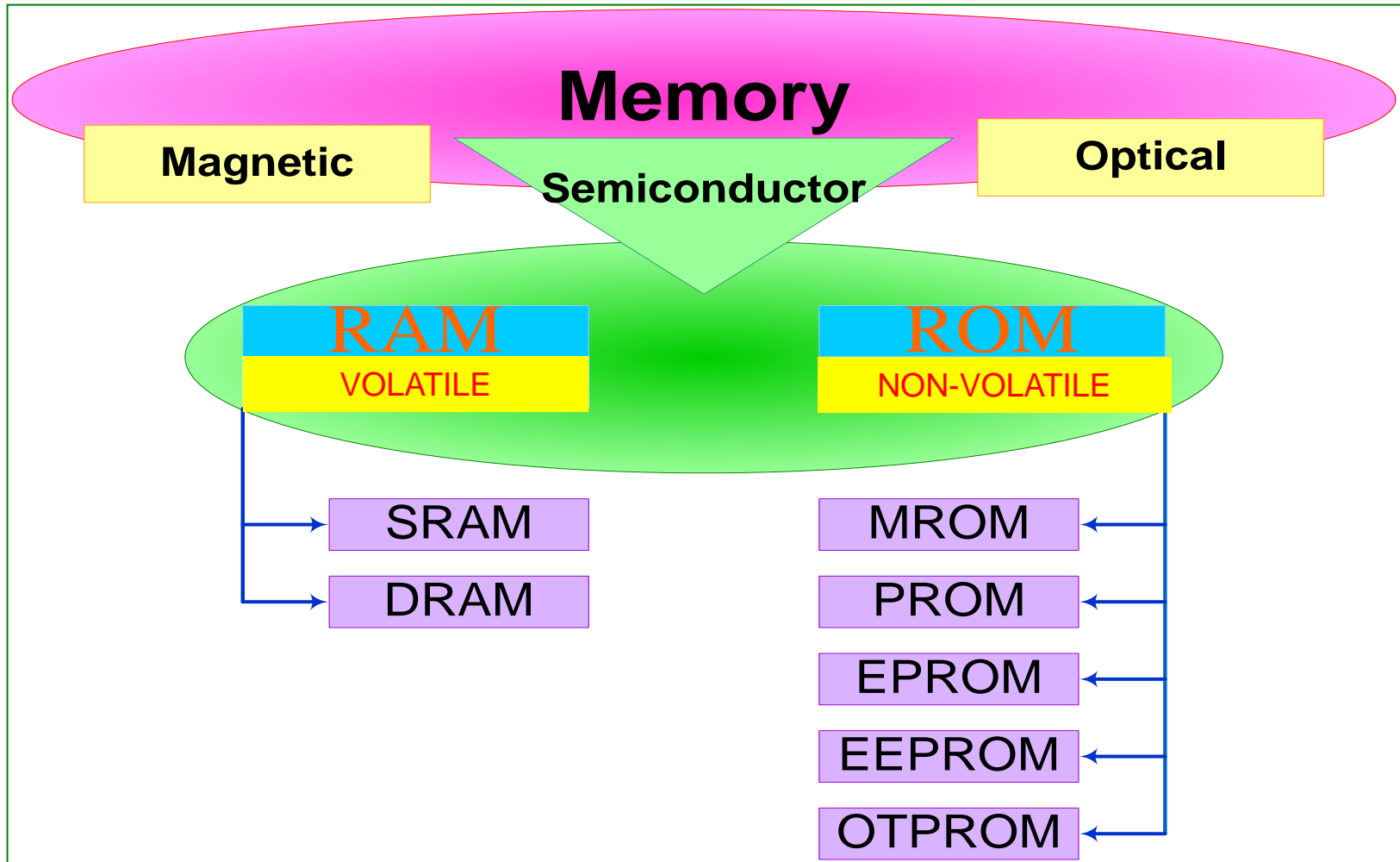
Επιλογή Ολοκληρωμένου
(Chip Select)



Έξοδοι Δεδομένων
(Data Outputs)



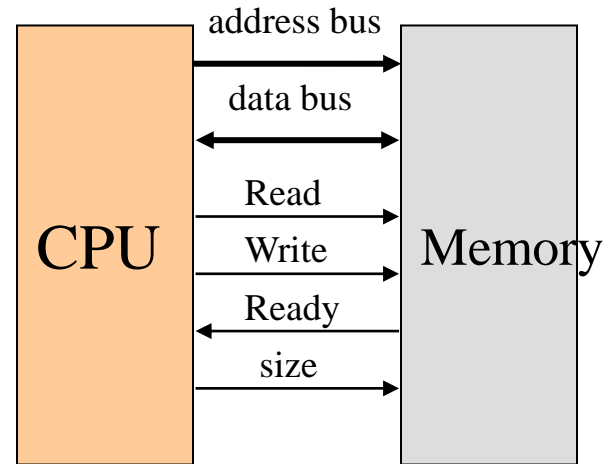
Memory Classification by technology



Generic CPU Memory Interface

□ CPU Memory Interface usually consists of:

- unidirectional address bus
- bidirectional data bus
- read control line
- write control line
- ready control line
- size (byte, word) control line



□ Memory access involves a memory bus transaction

■ read:

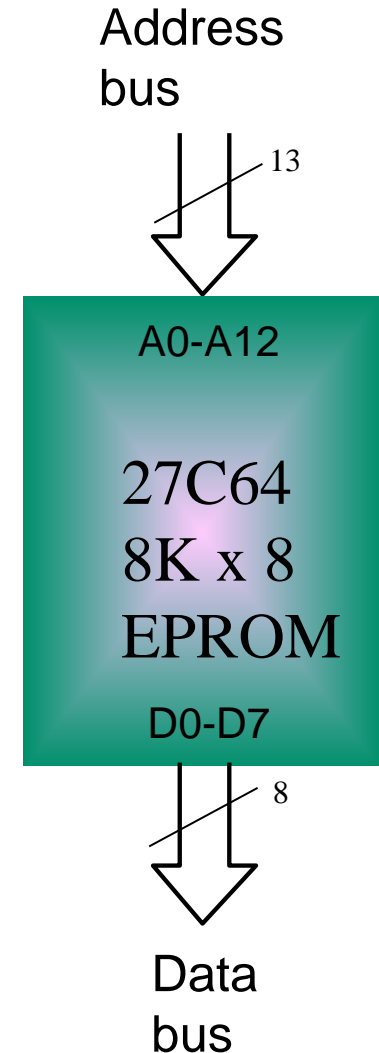
- (1) set address, read and size,
- (2) copy data when ready is set by memory

■ write:

- (1) set address, data, write and size,
- (2) done when ready is set

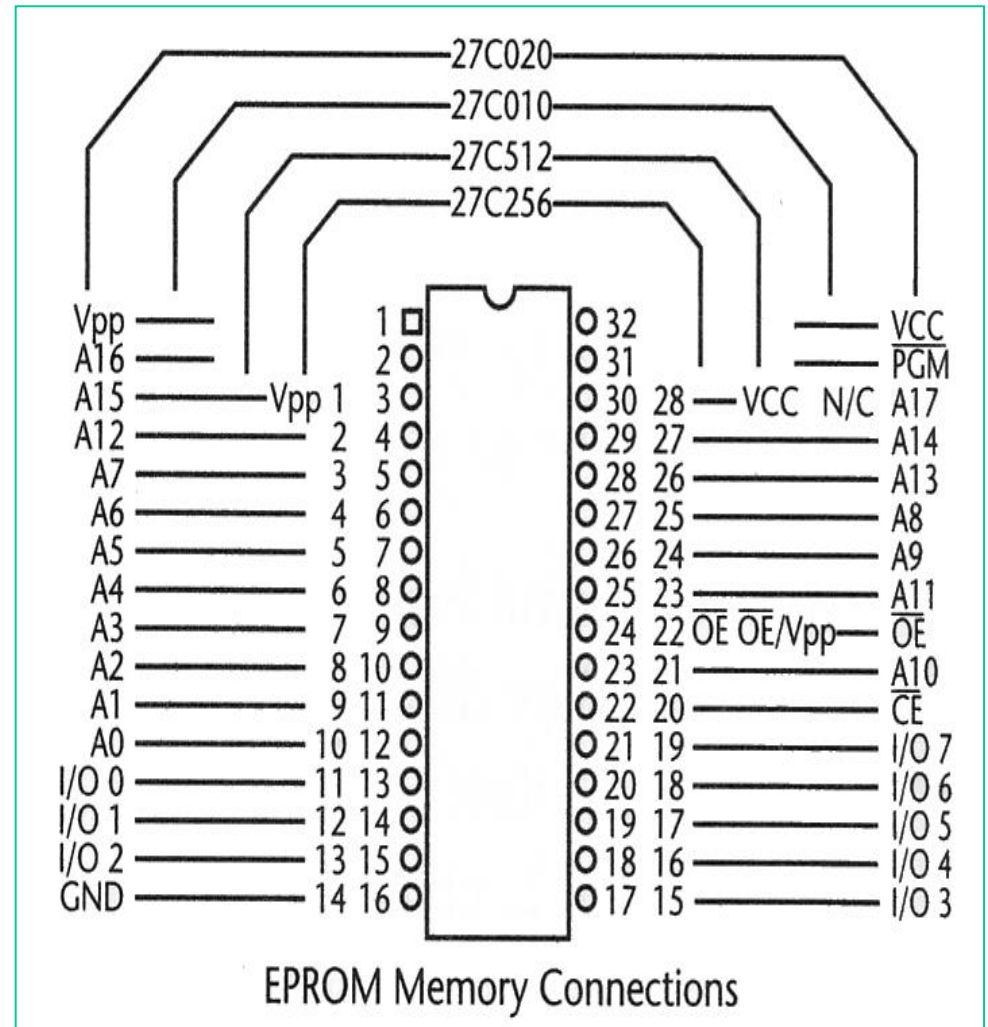
Memory Organization

- ❑ Number of address and data lines on a memory chip is related to how bits are arranged internally.
- ❑ Number of locations = $2^{\text{address lines}}$:
 - **Example:** 27C64 chip has 13 lines numbered from A0 through A12
 - It has $2^{13} = 2^3 \times 2^{10} = 8 \times \text{kilo} = 8192$ locations
- ❑ Number of bits per location = data lines:
 - **Example:** 27C64 chip has 8 data lines numbered from D0 through D7.
- ❑ Memory organization is a shorthand notation to describe all the information stated above.
- ❑ Memory organization is $2^{\text{address lines}} \times \text{data lines}$:
 - **Example:** 27C64 EPROM is has 8K x 8 organization.
 - 8 kilo locations x 8 = 8192 x 8 = 65536 bits total .

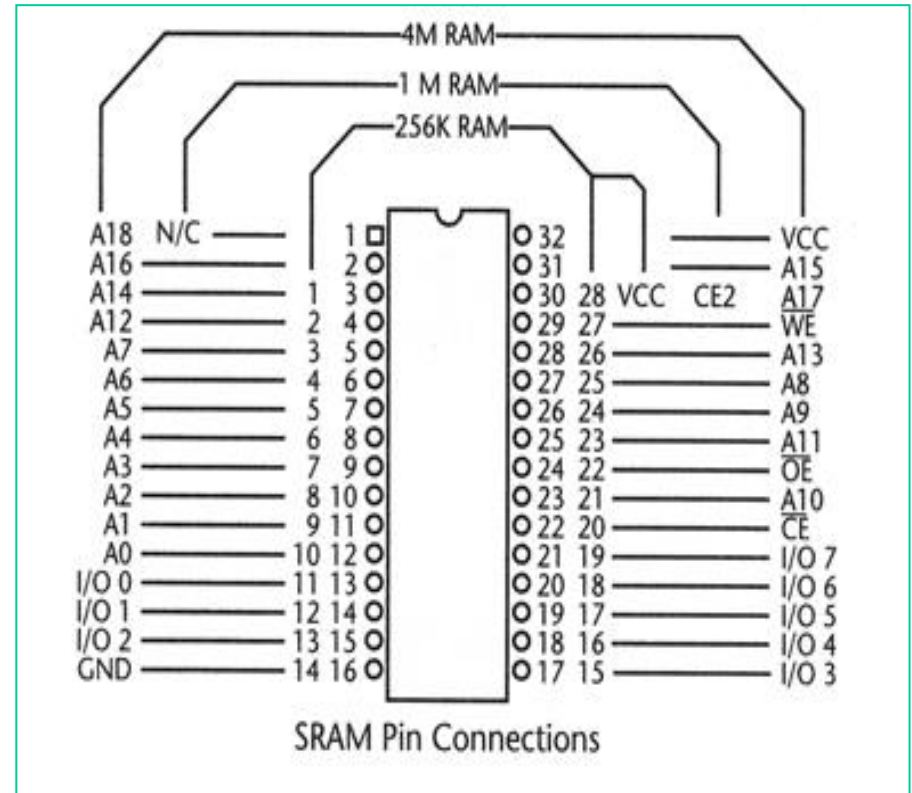
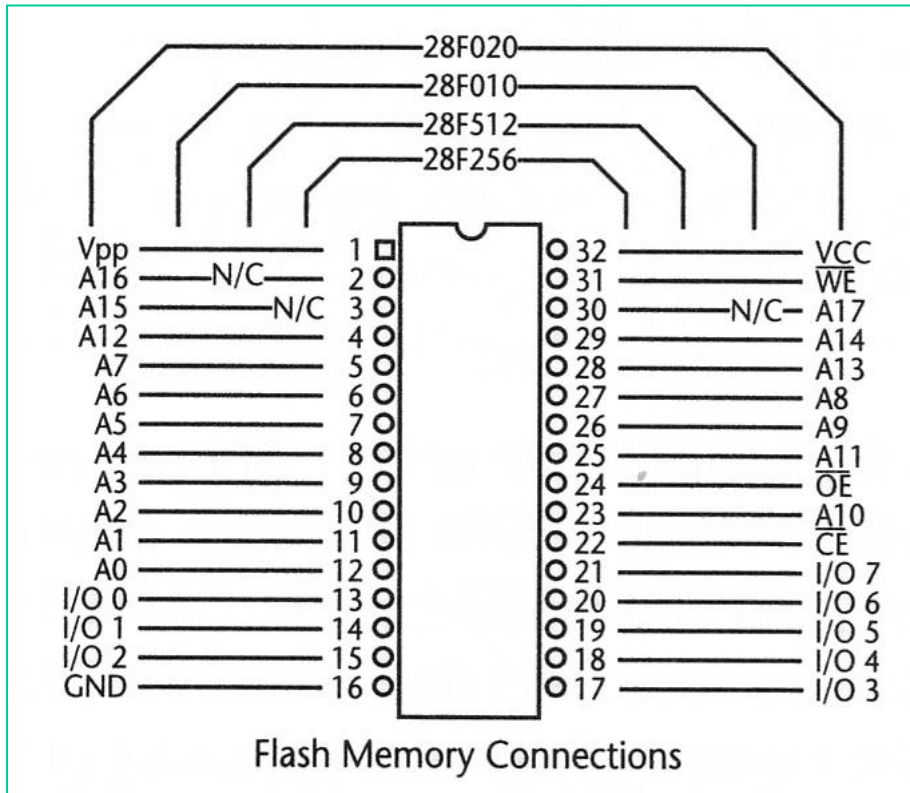


JEDEC

- ❖ Joint Electronic Device Engineering Council.
- ❖ Standard pinout for many related memories.
- ❖ Interchangeability of devices from different vendors.



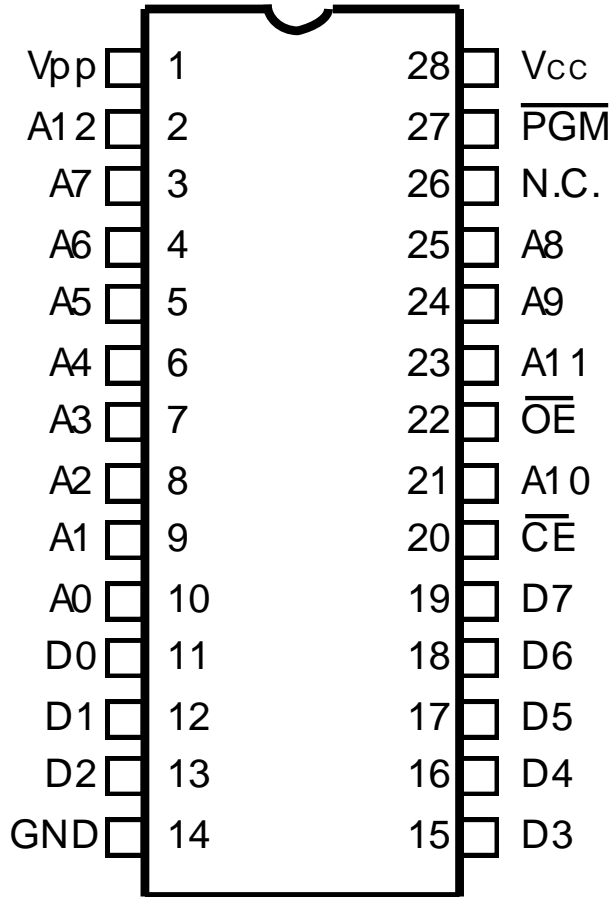
More JEDEC



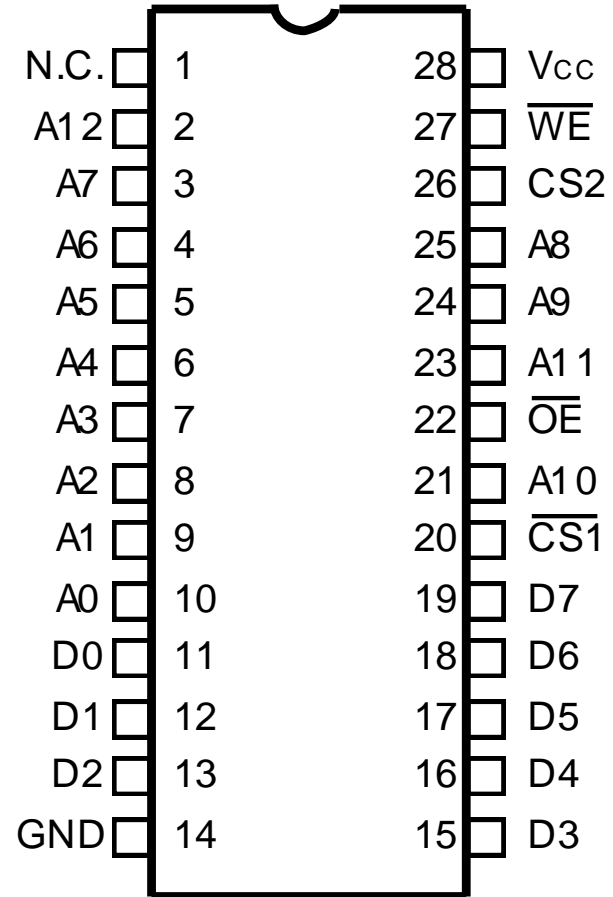
JEDEC Industry Standard

Pinouts

2764 EPROM



6264 SRAM



Part Numbers vs Capacity

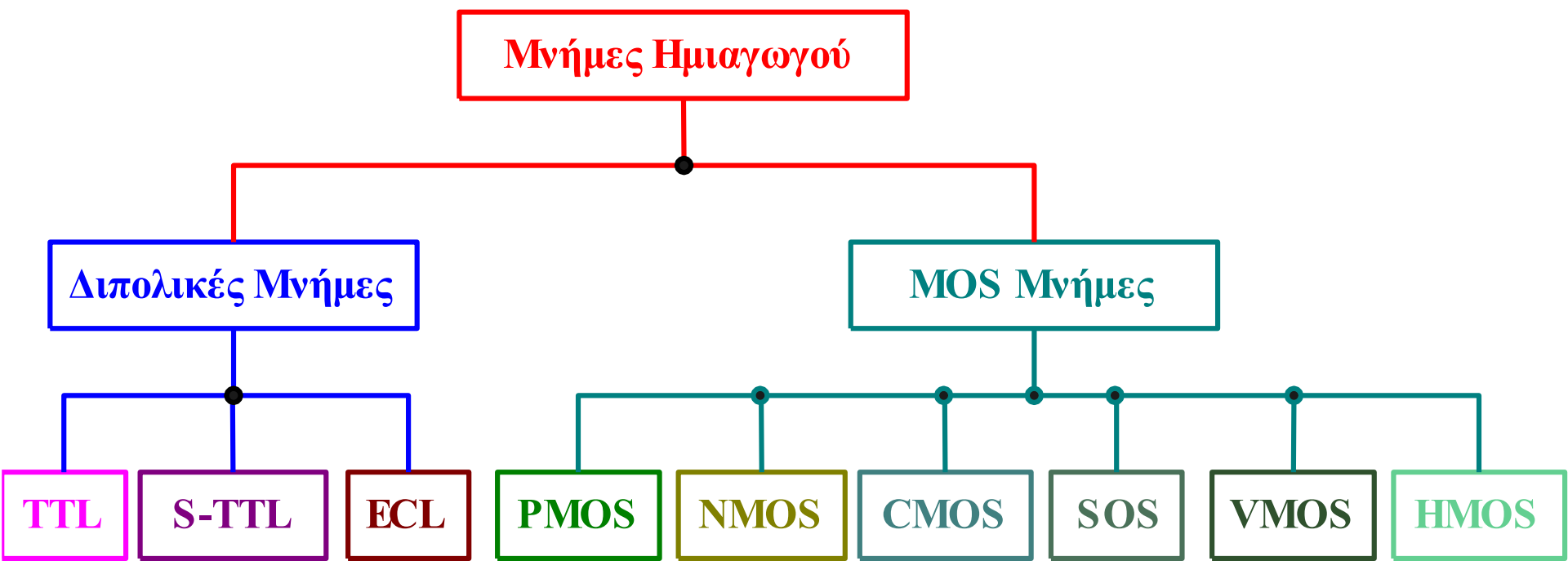
- It's very easy to derive memory capacity from the part number
 - From capacity (bits), divide by 8 to get number of addresses (byte locations).

Chip	Capacity	Organization	#addr lines
2732	32 kilobits	4k x 8	12
2764	64 kilobits	8k x 8	13
27128	128 kilobits	16k x 8	14
27256	256 kilobits	32k x 8	15
27512	512 kilobits	64k x 8	16
27C010	1 Megabit	128k x 8	17
27C020	2 Megabit	256k x 8	18
27C040	4 Megabits	512k x 8	19

Τεχνολογίες ημιαγωγού μνήμης

Οι μνήμες ημιαγωγού κατατάσσονται, όσον αφορά την επίδοση, σύμφωνα με τους παρακάτω συντελεστές:

- Την πυκνότητα ψηφίων ανά ολοκληρωμένο.
- Την ταχύτητα (χρόνος προσπέλασης).
- Την κατανάλωση ισχύος.
- Το κόστος ανά αποθηκευμένο ψηφίο.
- Το βαθμό ανοσίας στο θόρυβο.



Τα βασικά χαρακτηριστικά της μνήμης TTL είναι:

- Υψηλές ταχύτητες.
- Μέτρια χωρητικότητα.
- Υψηλή κατανάλωση ισχύος.
- Σχετικά χαμηλός βαθμός ανοσίας στο θόρυβο.
- Σχετικά υψηλό κόστος.

Η μνήμη MOS είναι κατασκευασμένη κυρίως με τεχνολογία NMOS.

Τα κύρια χαρακτηριστικά αυτής της μνήμης είναι:

- Η υψηλότερη χωρητικότητα από όλες τις τεχνολογίες μνήμης.
- Το χαμηλότερο κόστος ανά ψηφίο.
- Η πολύ χαμηλότερη κατανάλωση ισχύος απ' ό,τι η διπολική μνήμη.

- Το μόνο της μειονέκτημα είναι η χαμηλότερη ταχύτητα συγκριτικά με τη διπολική μνήμη, μειονέκτημα που όλο και περισσότερο τείνει να εκλείψει, καθώς η ανάπτυξη της τεχνολογίας όλο και περισσότερο συμβάλλει στην αύξηση της ταχύτητας, έτσι που να κοντεύει να φτάσει την ταχύτητα της πιο αργής διπολικής μνήμης.
- Οι τεχνολογίες VMOS (Vertical MOS) και HMOS (High Performance MOS) έχουν πλησιάσει τις επιδόσεις της διπολικής τεχνολογίας.

Η τεχνολογία CMOS (Complementary MOS) έχει:

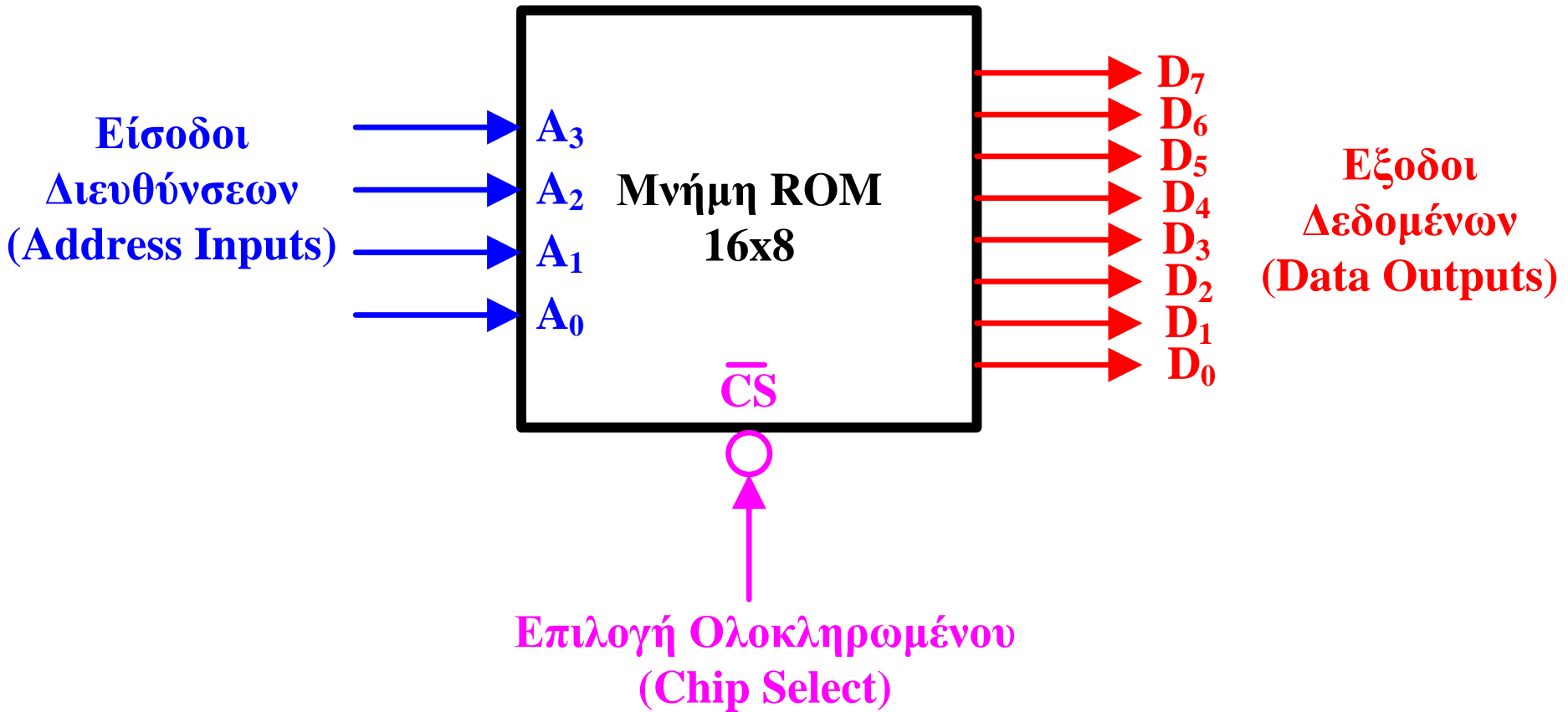
- Τη χαμηλότερη κατανάλωση ισχύος.
- Τον υψηλότερο βαθμό ανοσίας στο θόρυβο.
- Τη χαμηλότερη ταχύτητα από οποιαδήποτε άλλη τεχνολογία.
- Η χωρητικότητα είναι λίγο χαμηλότερη απ' ό,τι η NMOS αλλά υψηλότερη απ' ό,τι στη διπολική μνήμη.
- Μια παραλλαγή της τεχνολογίας CMOS, η SOS (Silicon on Sapphire), έχει επιτύχει μεγαλύτερες ταχύτητες και υψηλότερες πυκνότητες.

Μνήμη ROM

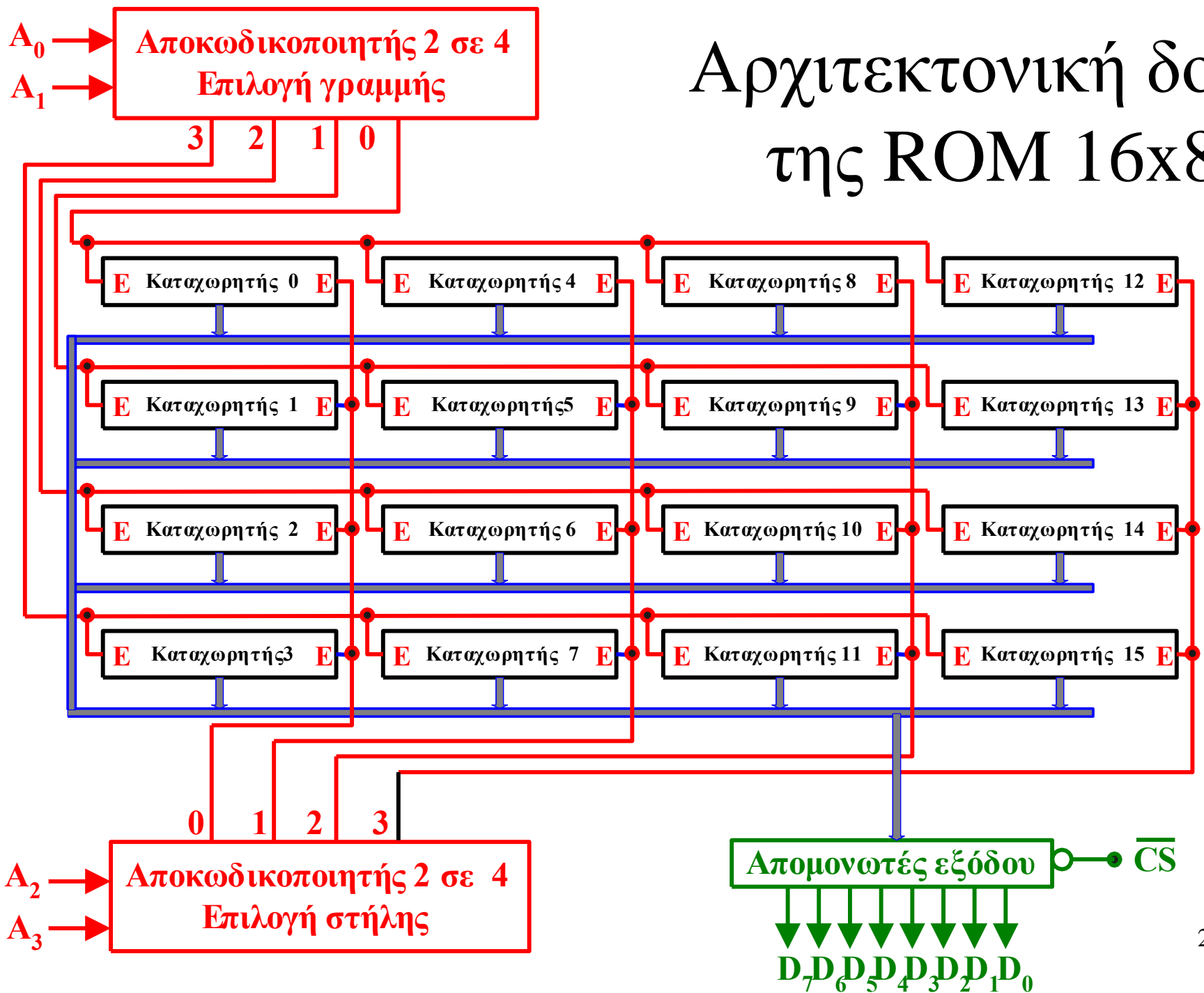
Read Only Memory (ROM)

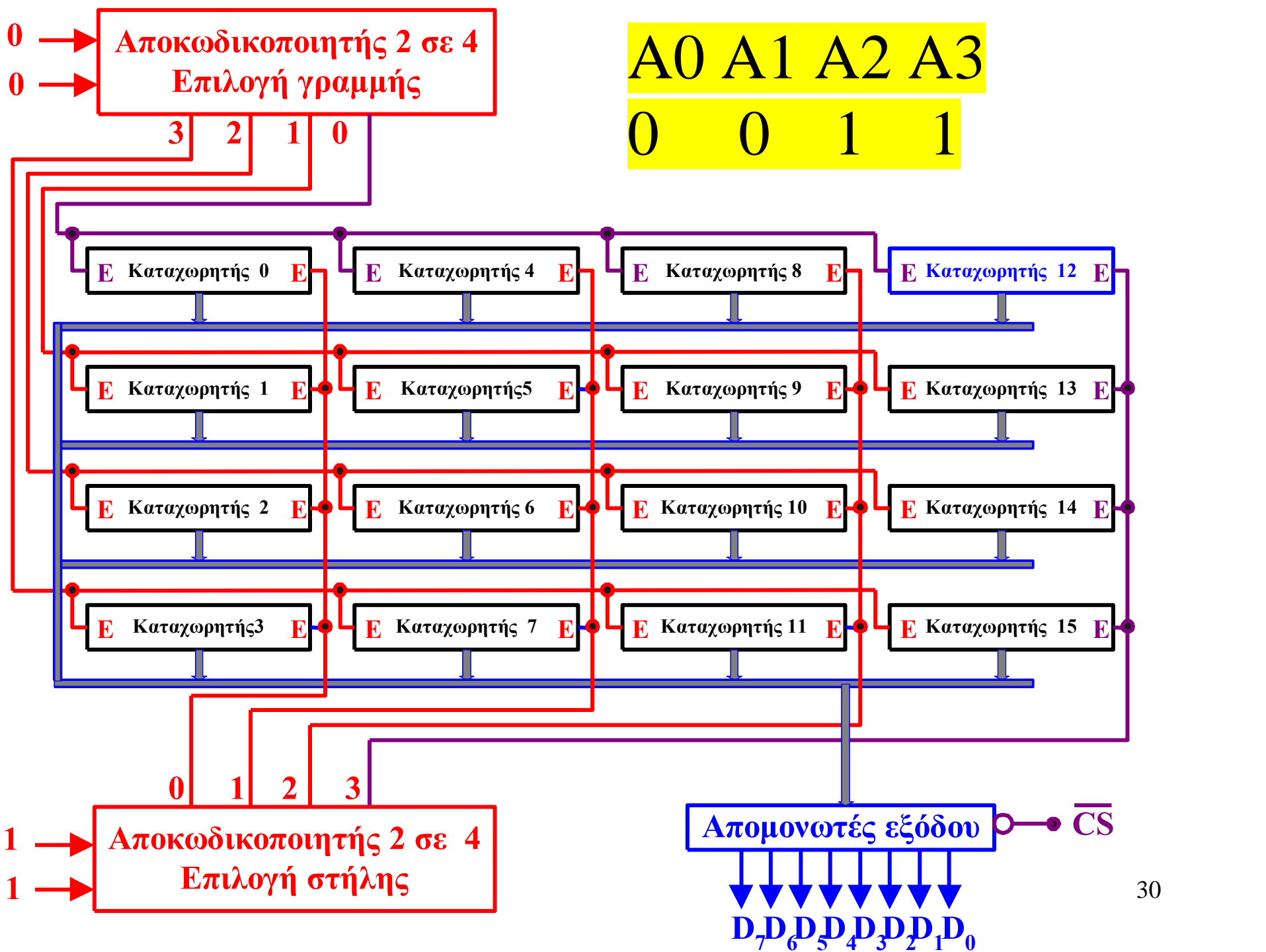
Μνήμη Ανάγνωσης Μόνο

- Μνήμη ανάγνωσης μόνον με μάσκα (**Masked Read Only Memory ή MRROM**).
- Προγραμματιζόμενη μνήμη ανάγνωσης μόνον (**Programmable ROM ή PROM**).
- Σβηνώμενη, προγραμματιζόμενη, μνήμη ανάγνωσης μόνο (**Erasable PROM ή EPROM**).
- Ηλεκτρικά αλλασσόμενη μνήμη ανάγνωσης μόνον (**Electrically Alterable ROM ή EAROM**).

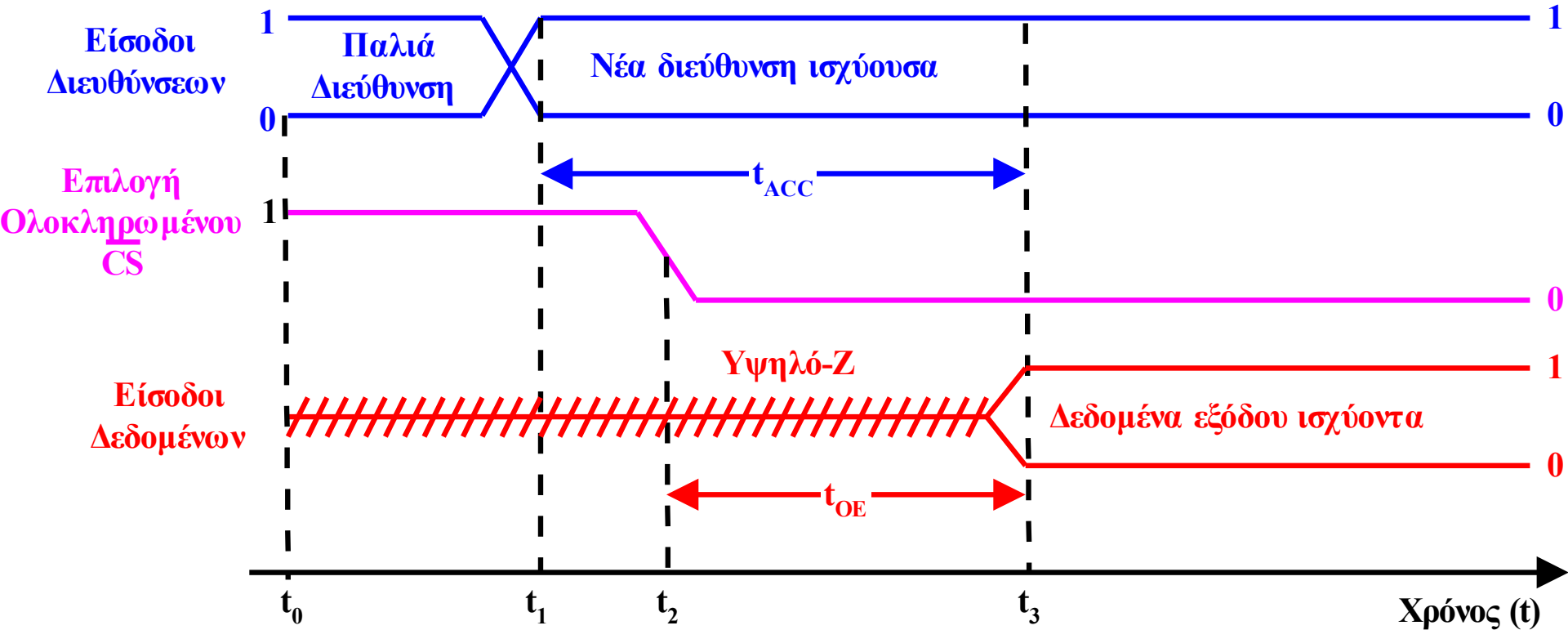


Αρχιτεκτονική δομή της ROM 16x8





Ανάγνωση από μια ROM



Μνήμη RAM

Random Access Memory (RAM)

Μνήμη τυχαίας προσπέλασης

Η μνήμη RAM χωρίζεται σε δύο μεγάλες κατηγορίες:

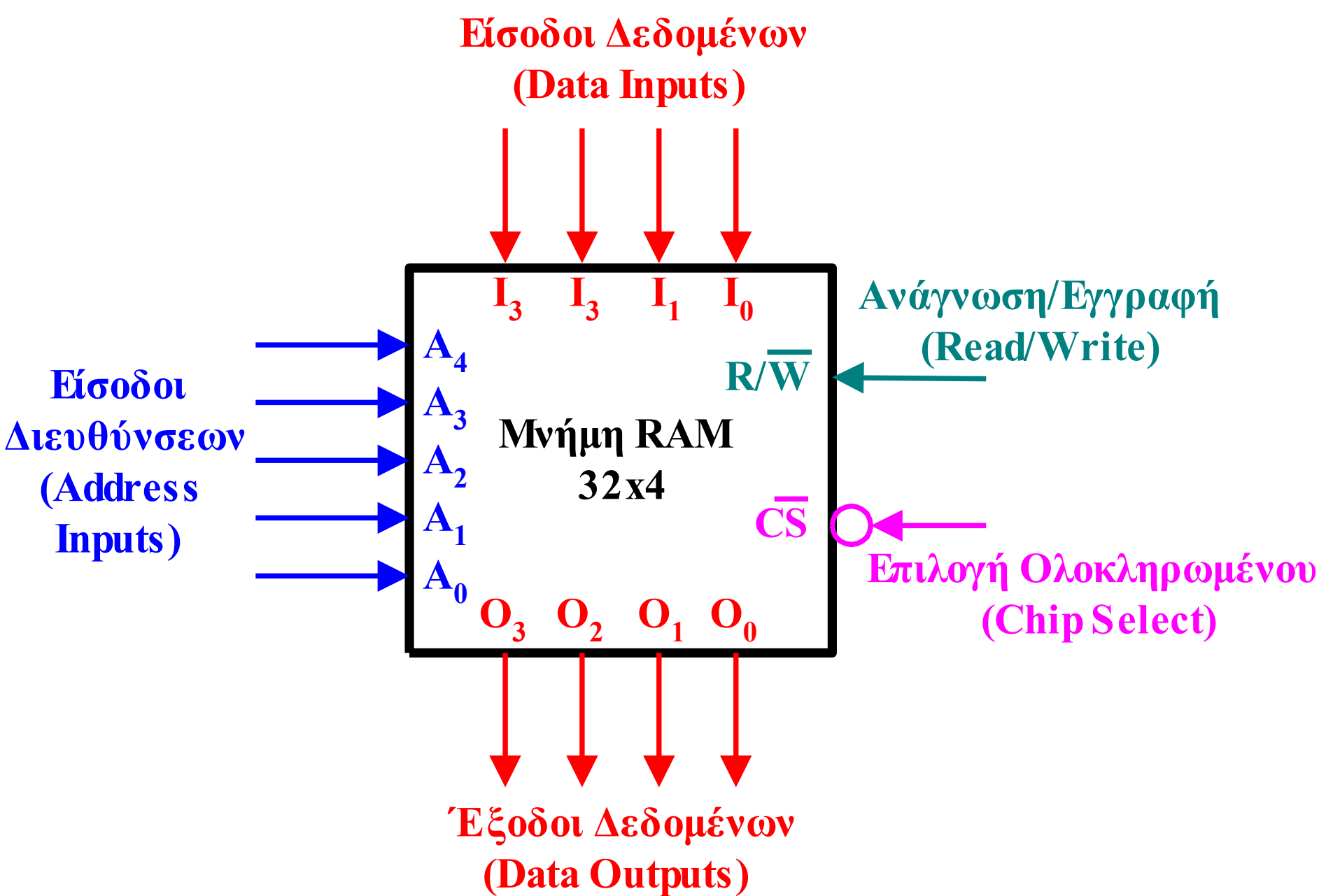
- Τη στατική μνήμη RAM (Static RAM ή SRAM)
- Τη δυναμική μνήμη RAM (Dynamic RAM ή DRAM)

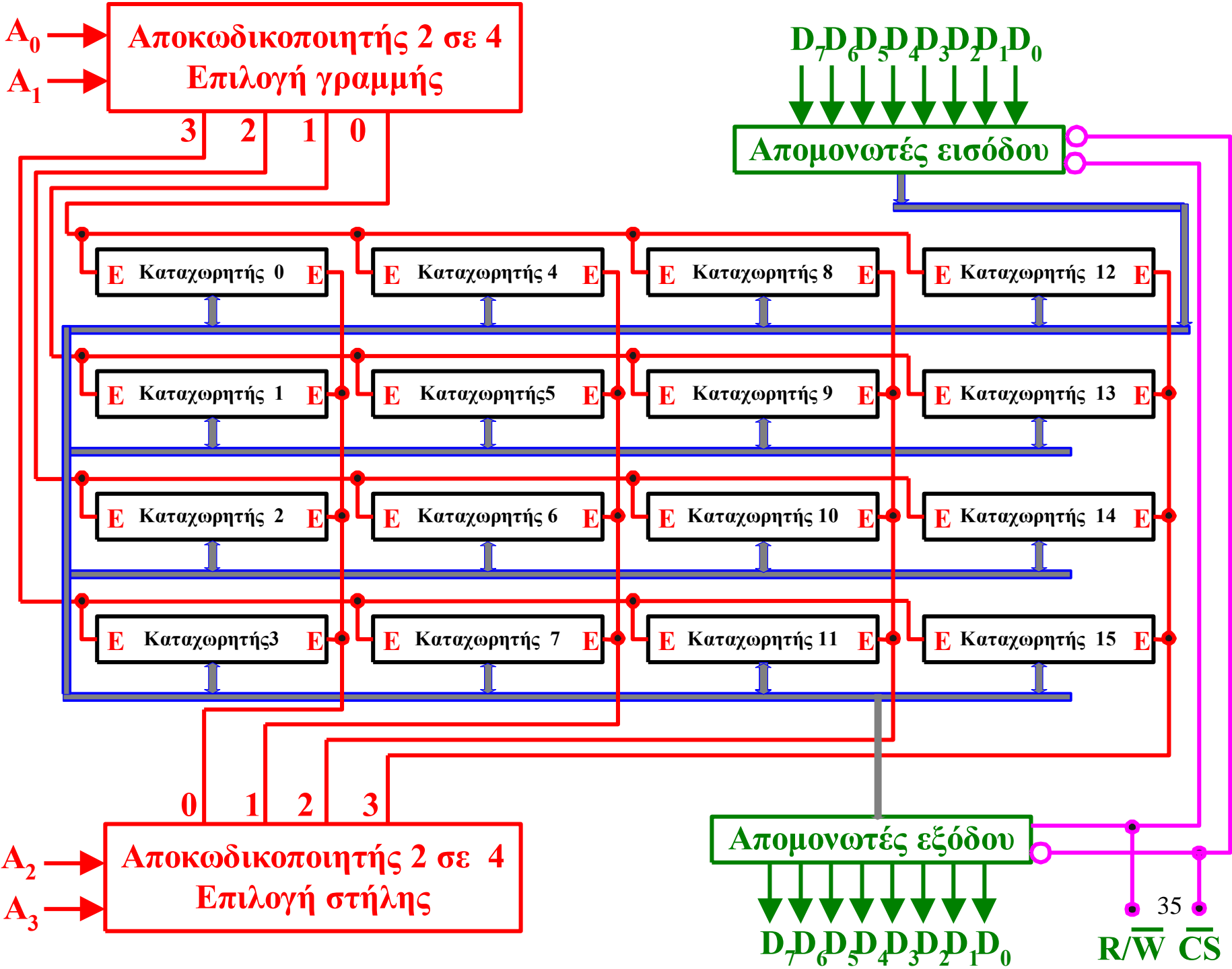
Η δυναμική RAM πλεονεκτεί σε σχέση με τη στατική RAM στο ότι:

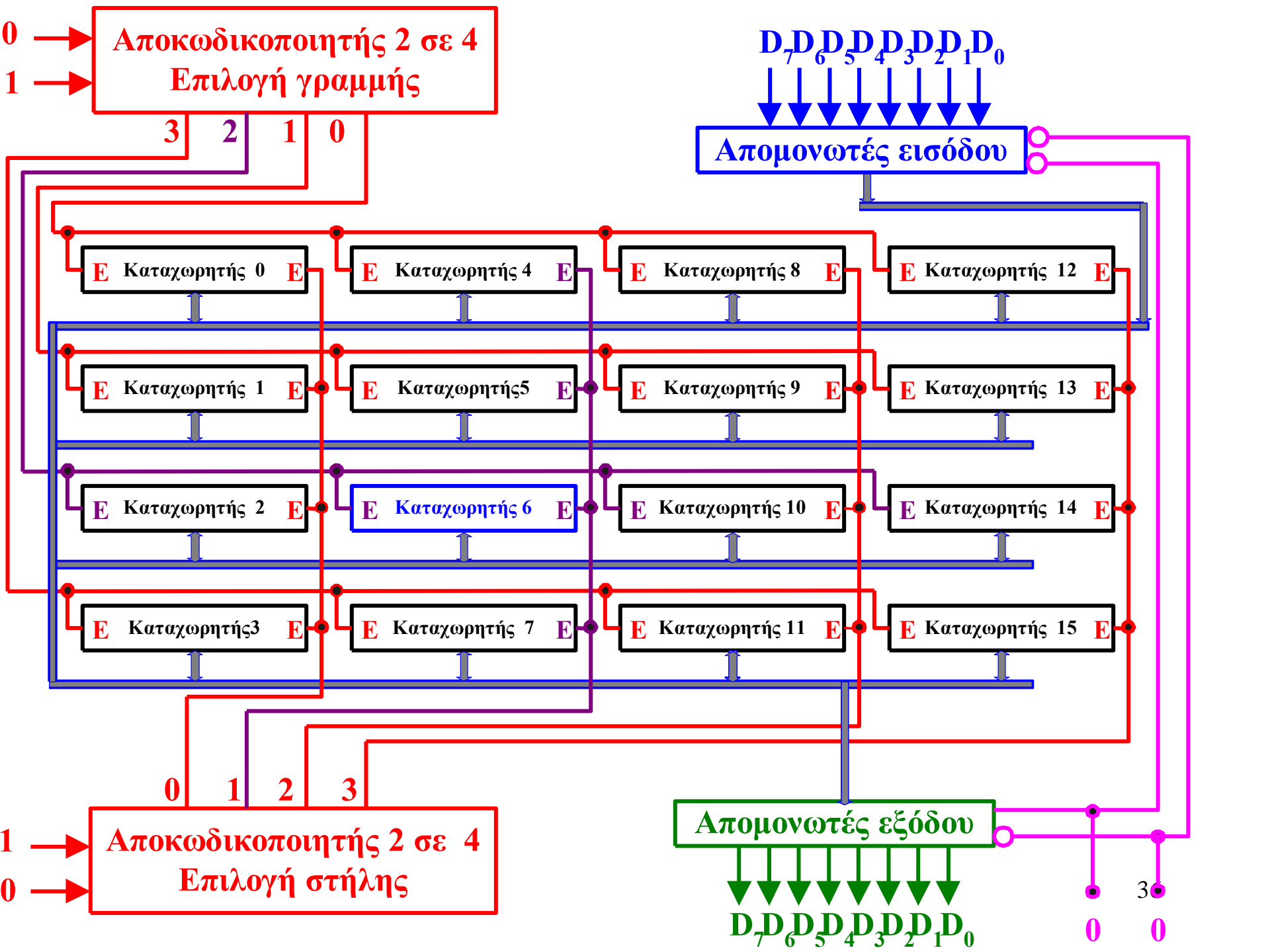
- Καταναλώνει λιγότερη ισχύ.
- Έχει μεγαλύτερη χωρητικότητα μνήμης.
- Έχει μικρότερο κόστος.

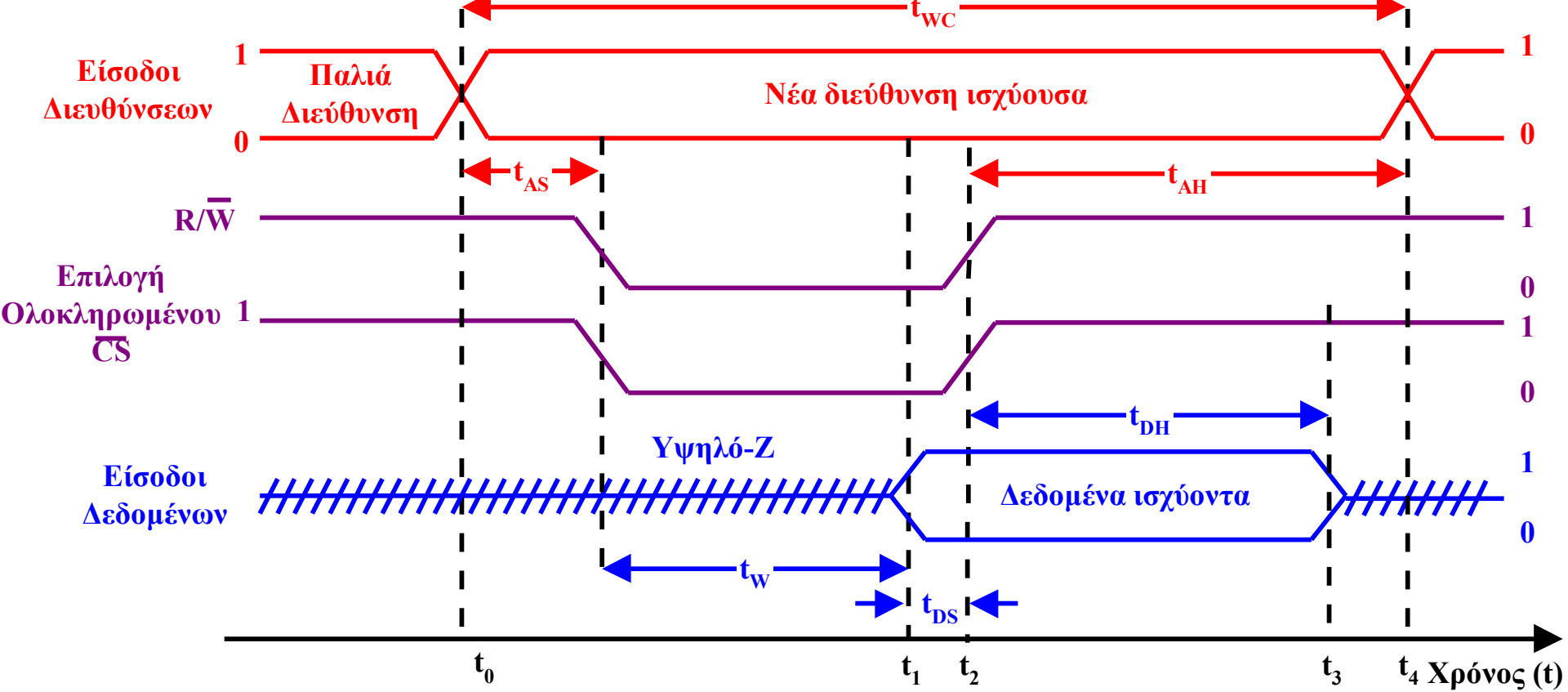
Η δυναμική RAM μειονεκτεί σε σχέση με τη στατική RAM στο ότι:

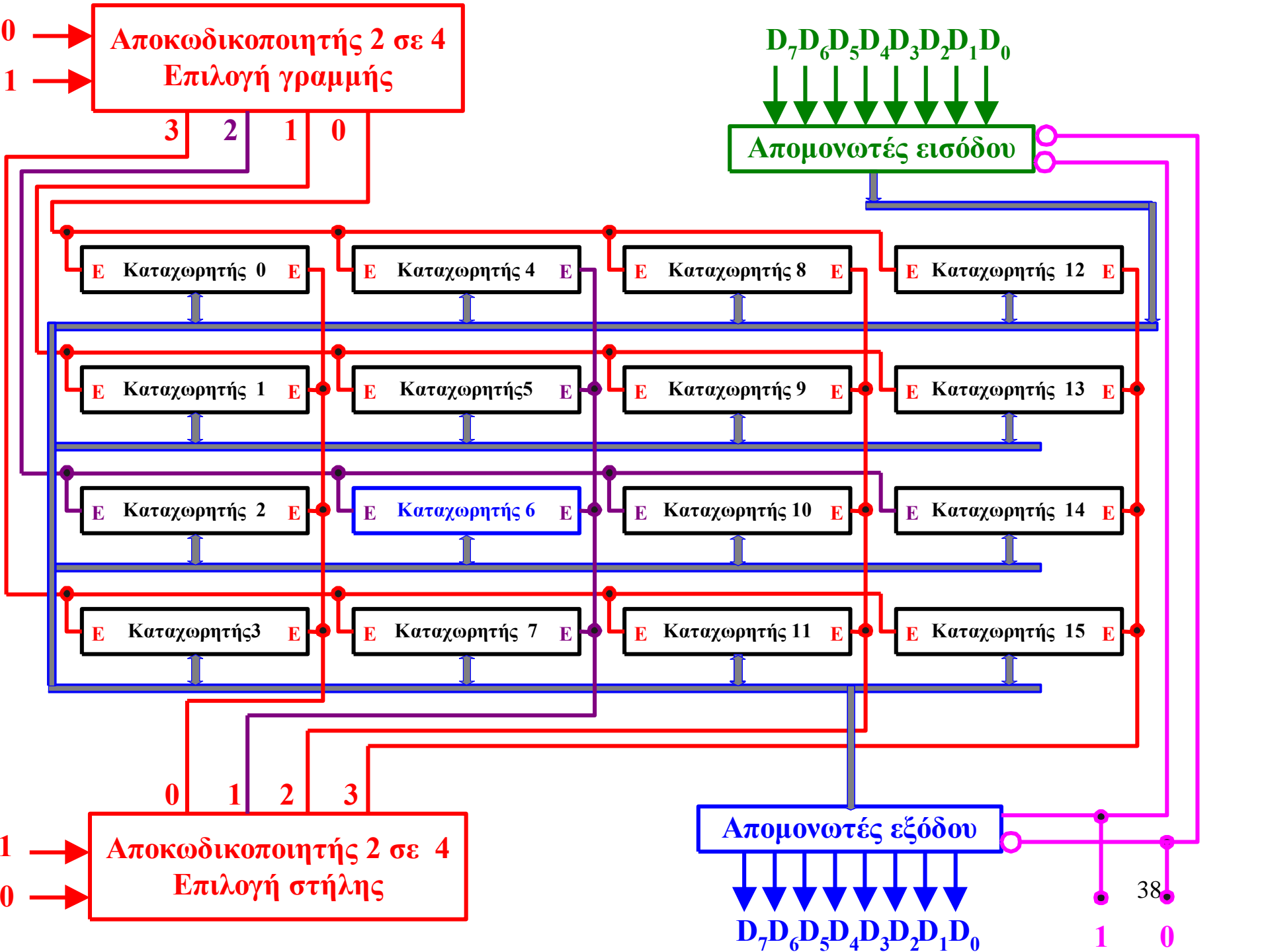
- Χρειάζεται περισσότερες από μία τάσεις τροφοδοσίας.
- Χρειάζεται επιπλέον κυκλώματα που θα εκτελέσουν τις λειτουργίες της πολύπλεξης διευθύνσεων (Address Multiplexing) και του φρεσκαρίσματος.
- Είναι σαφώς αργότερη.

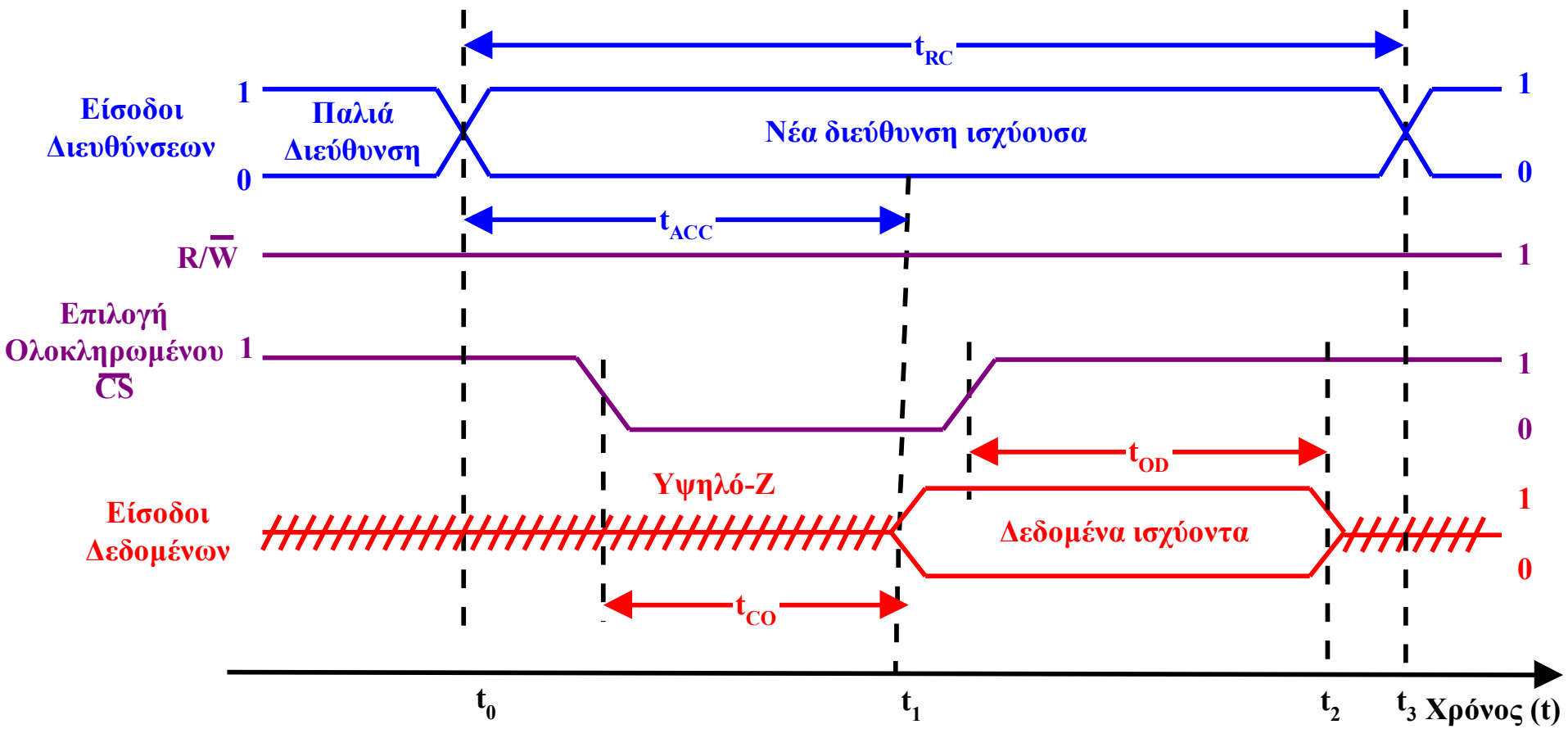


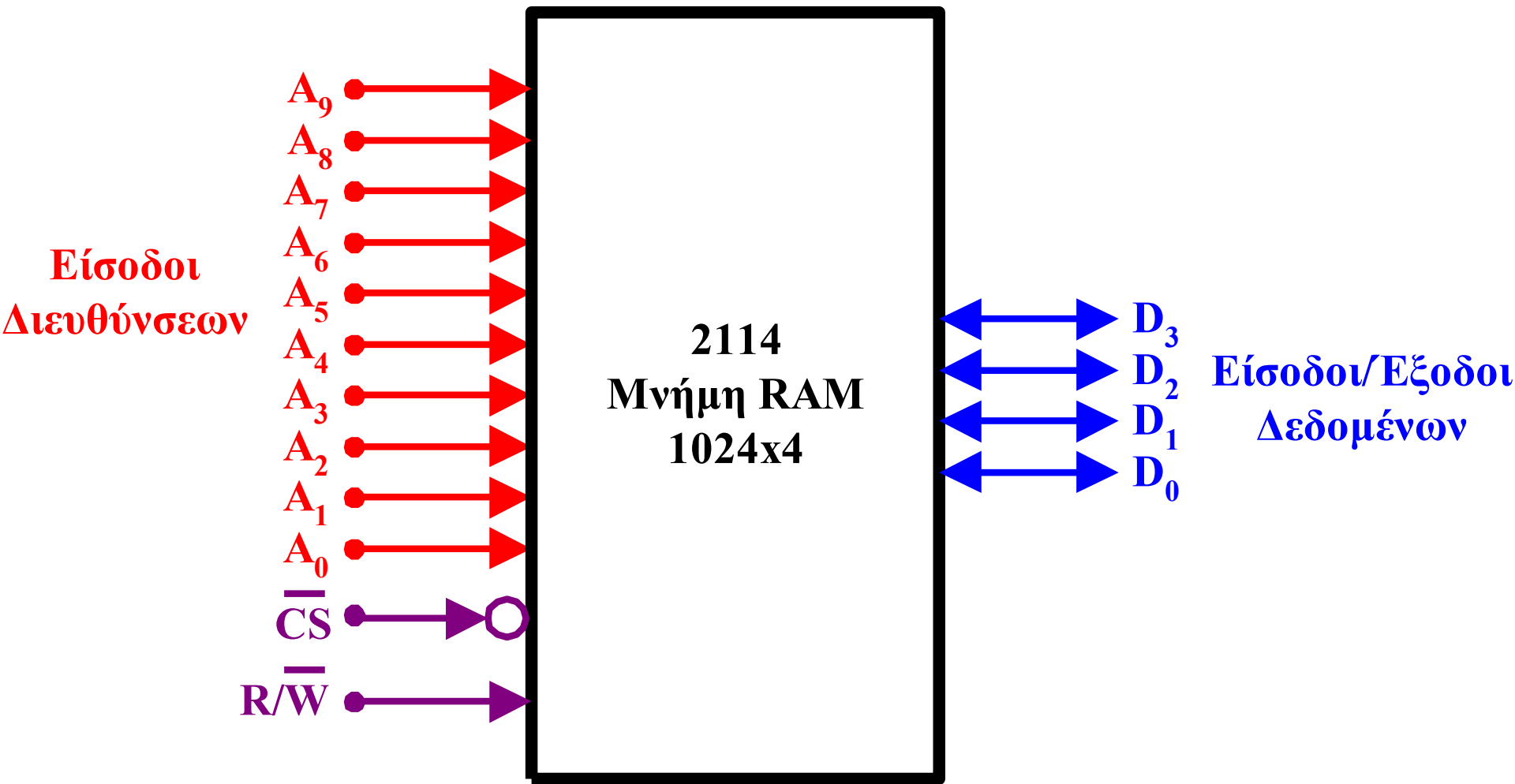




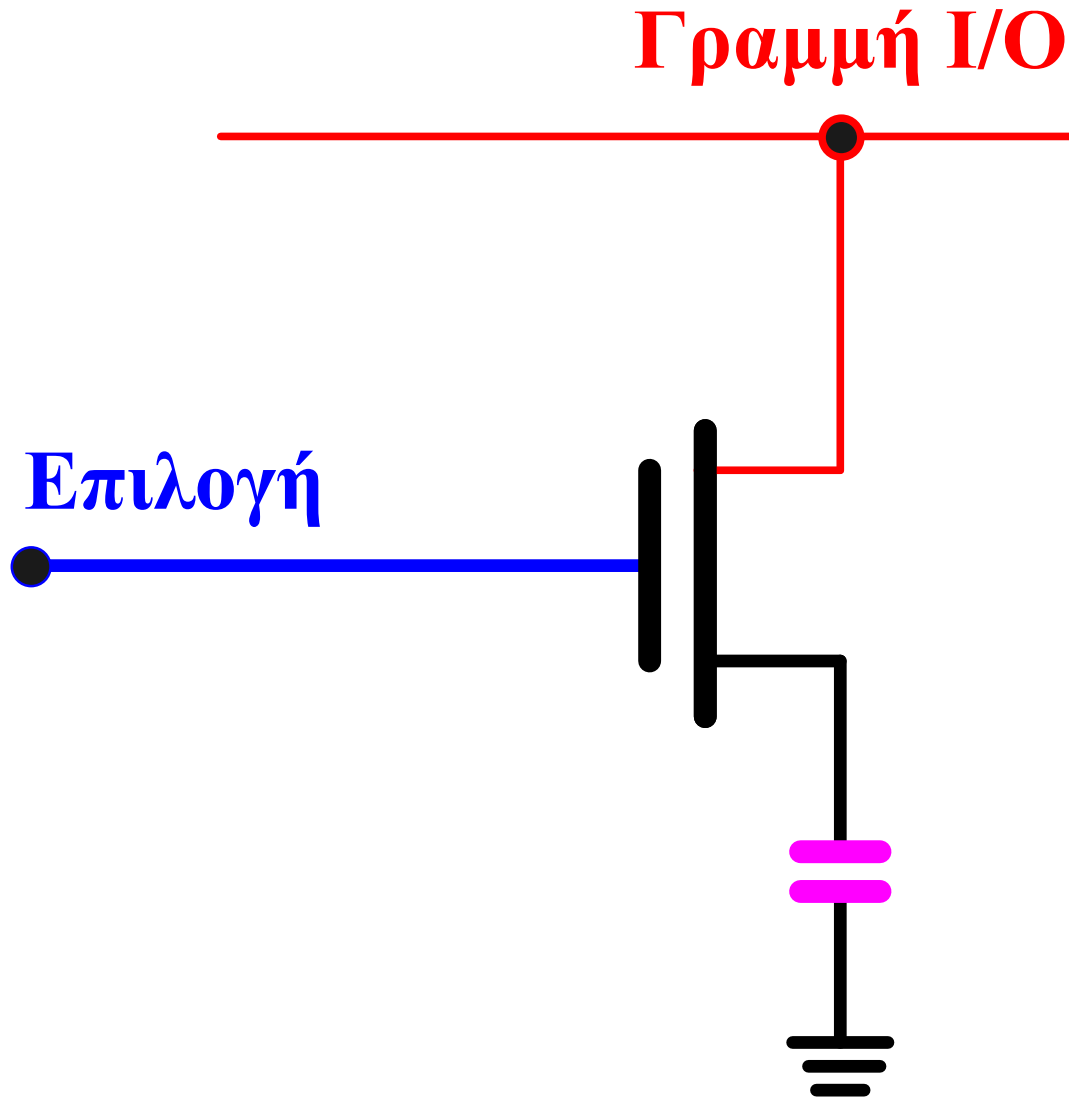


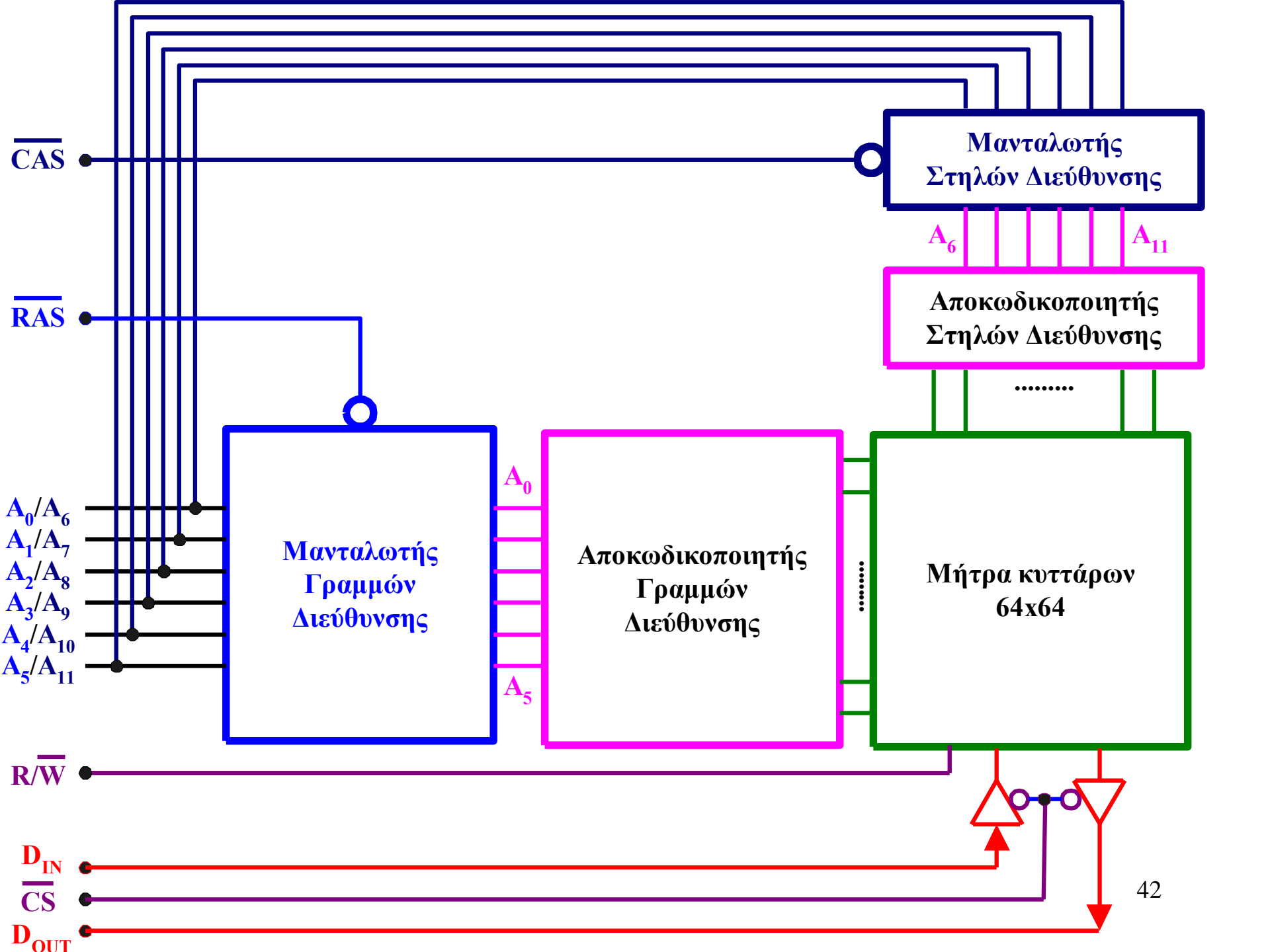


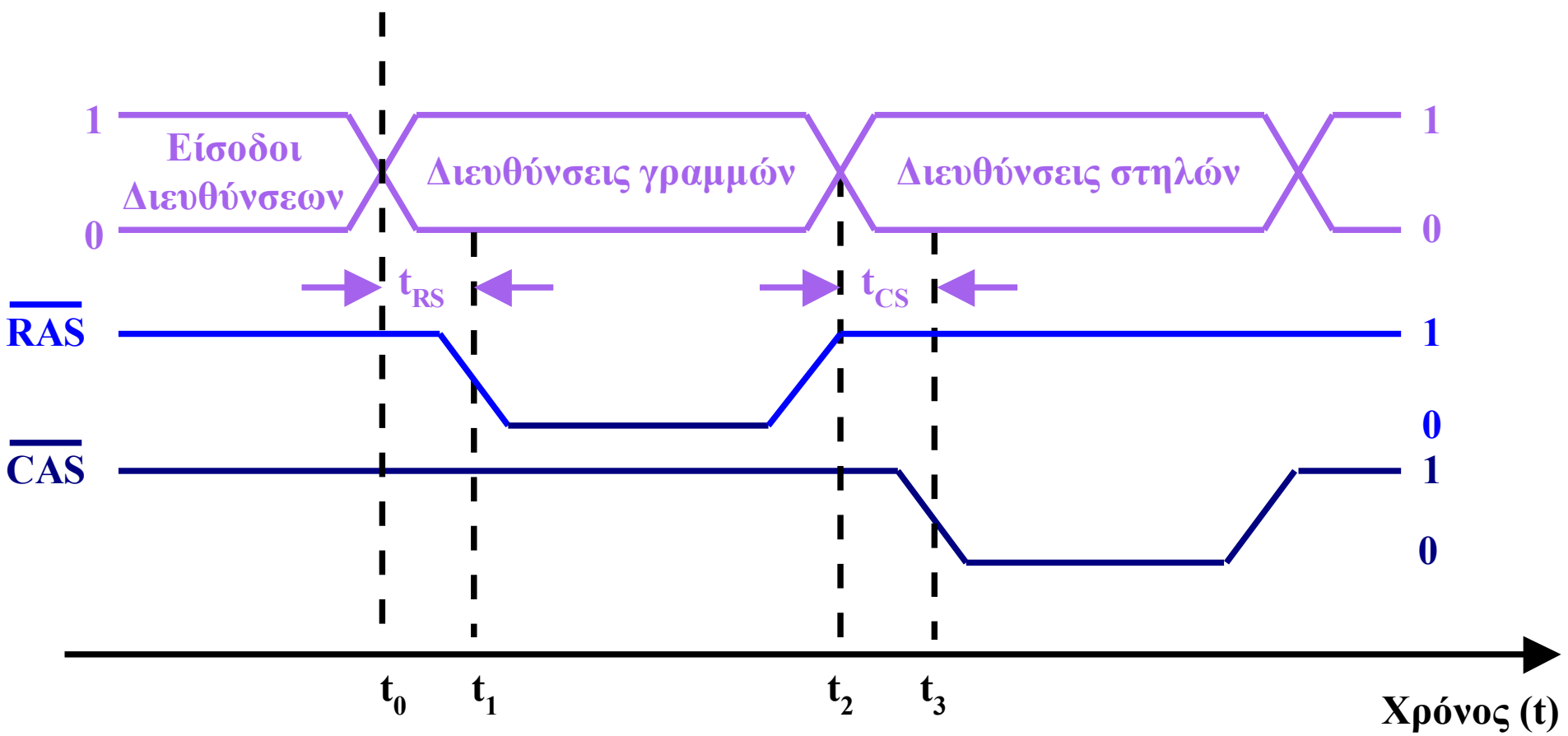




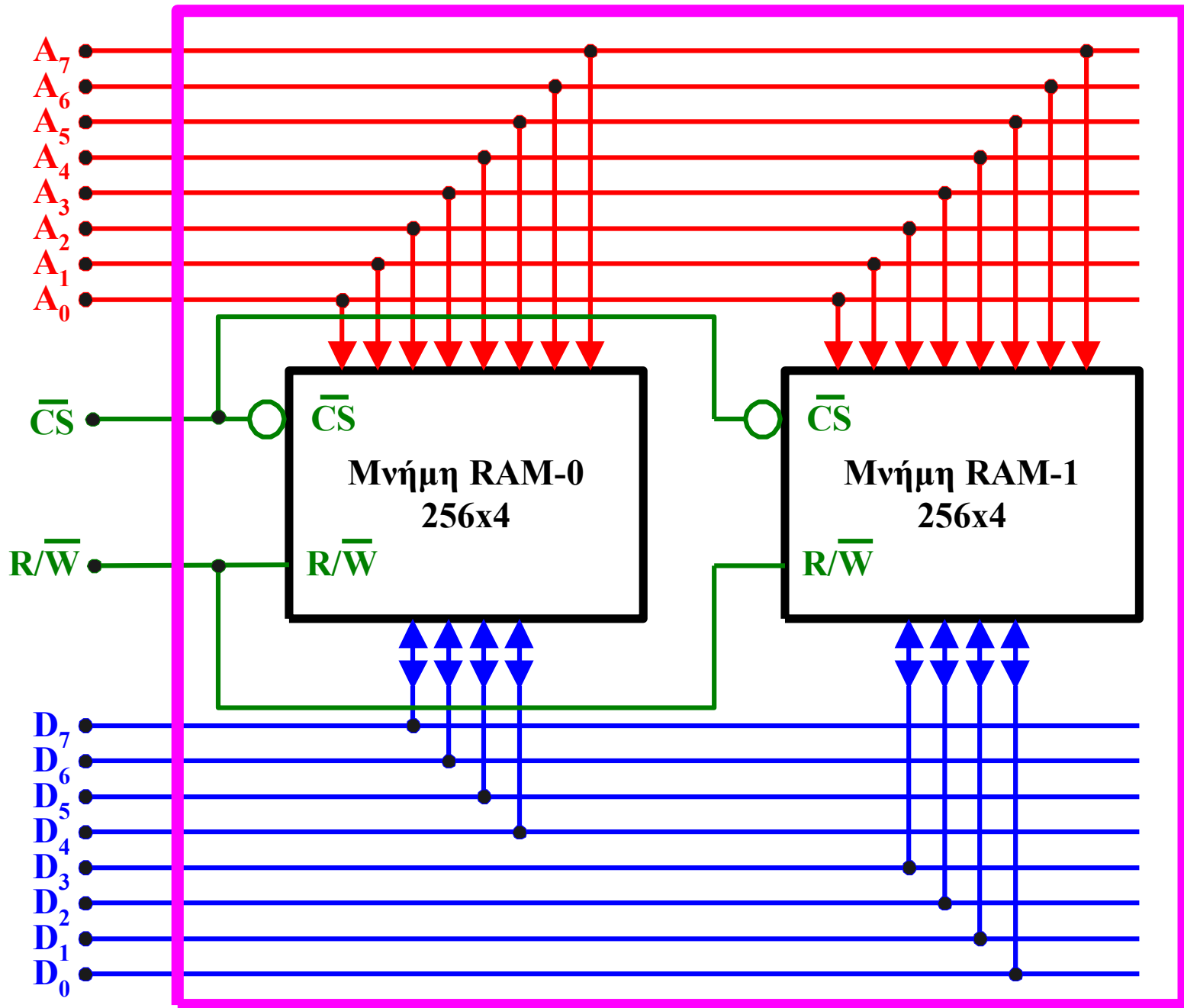
Δυναμική RAM



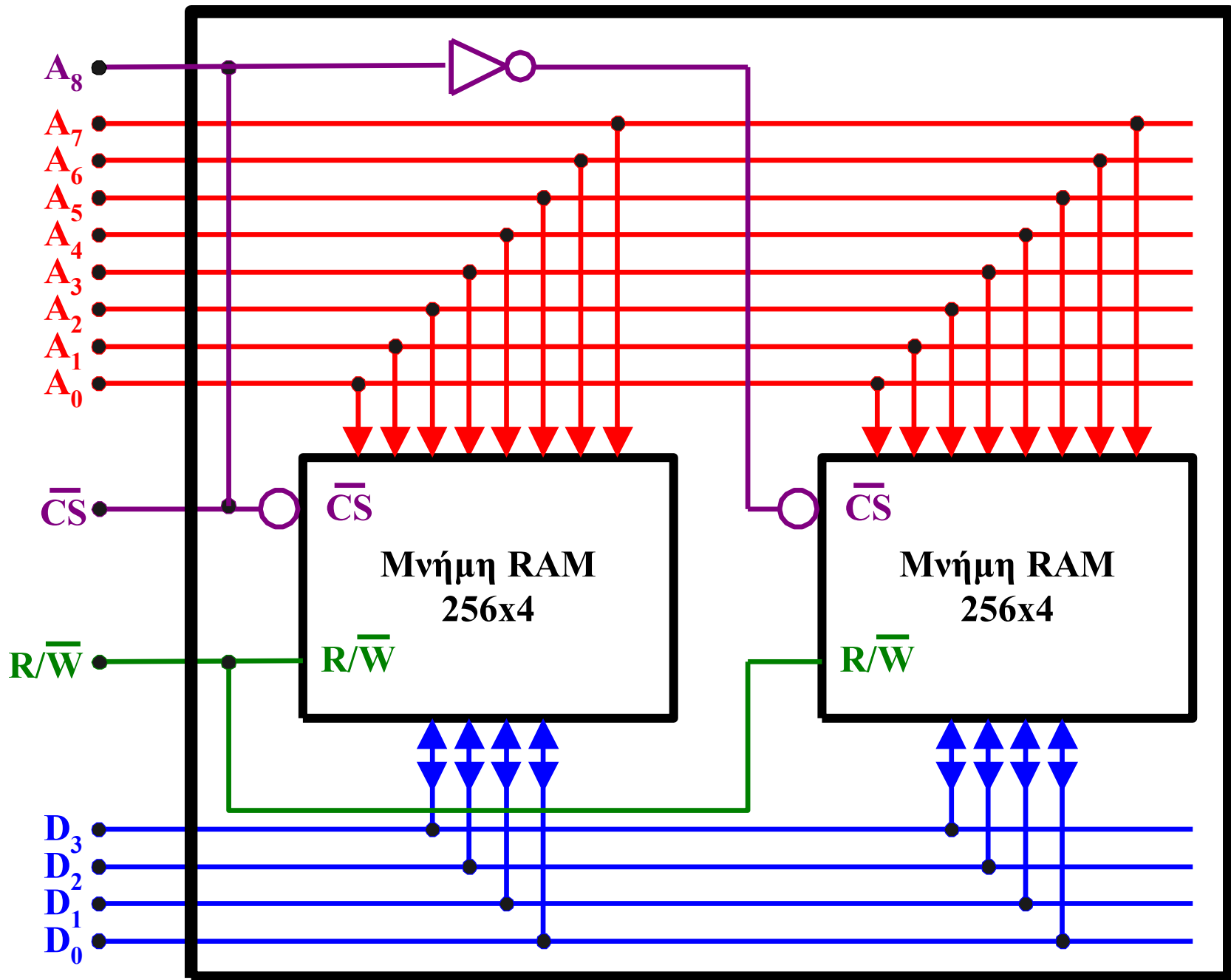




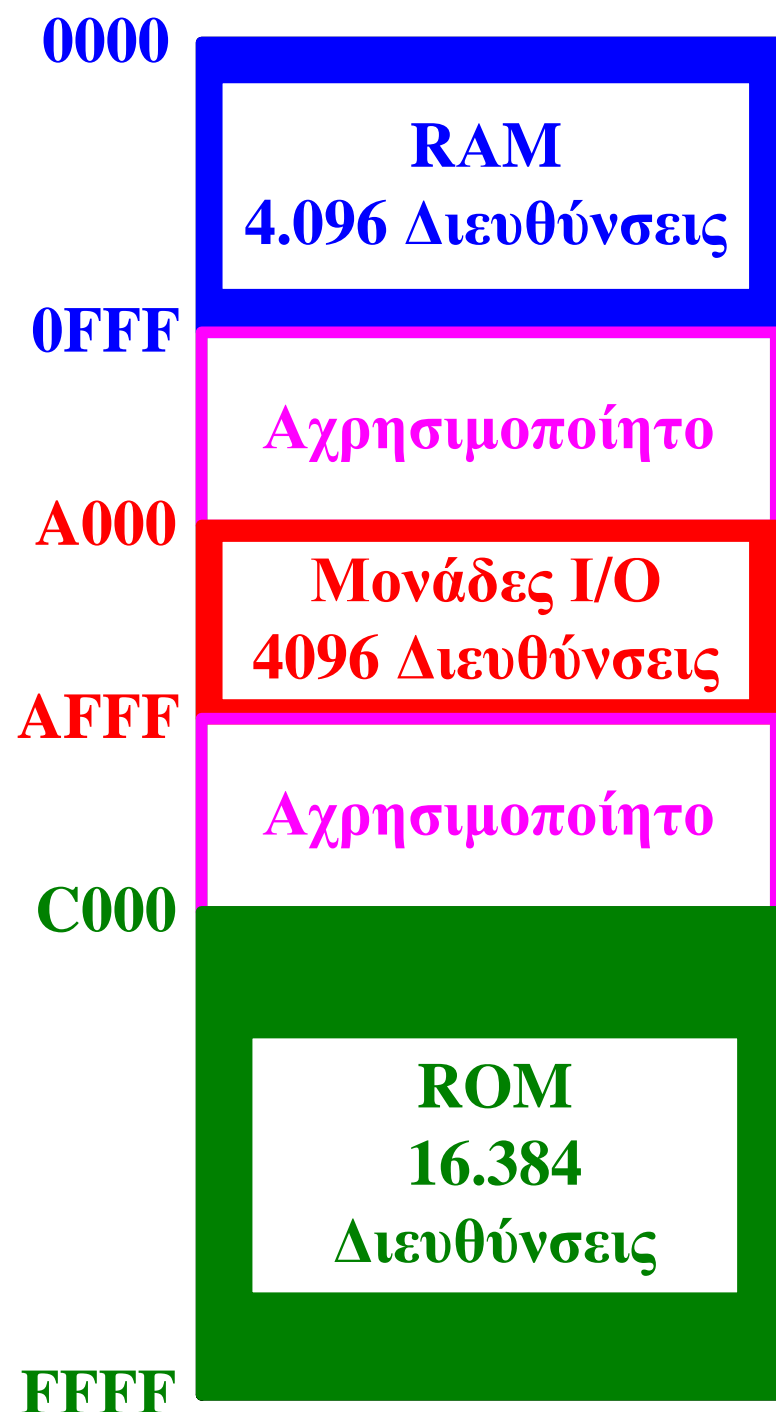
Επέκταση μήκους λέξης



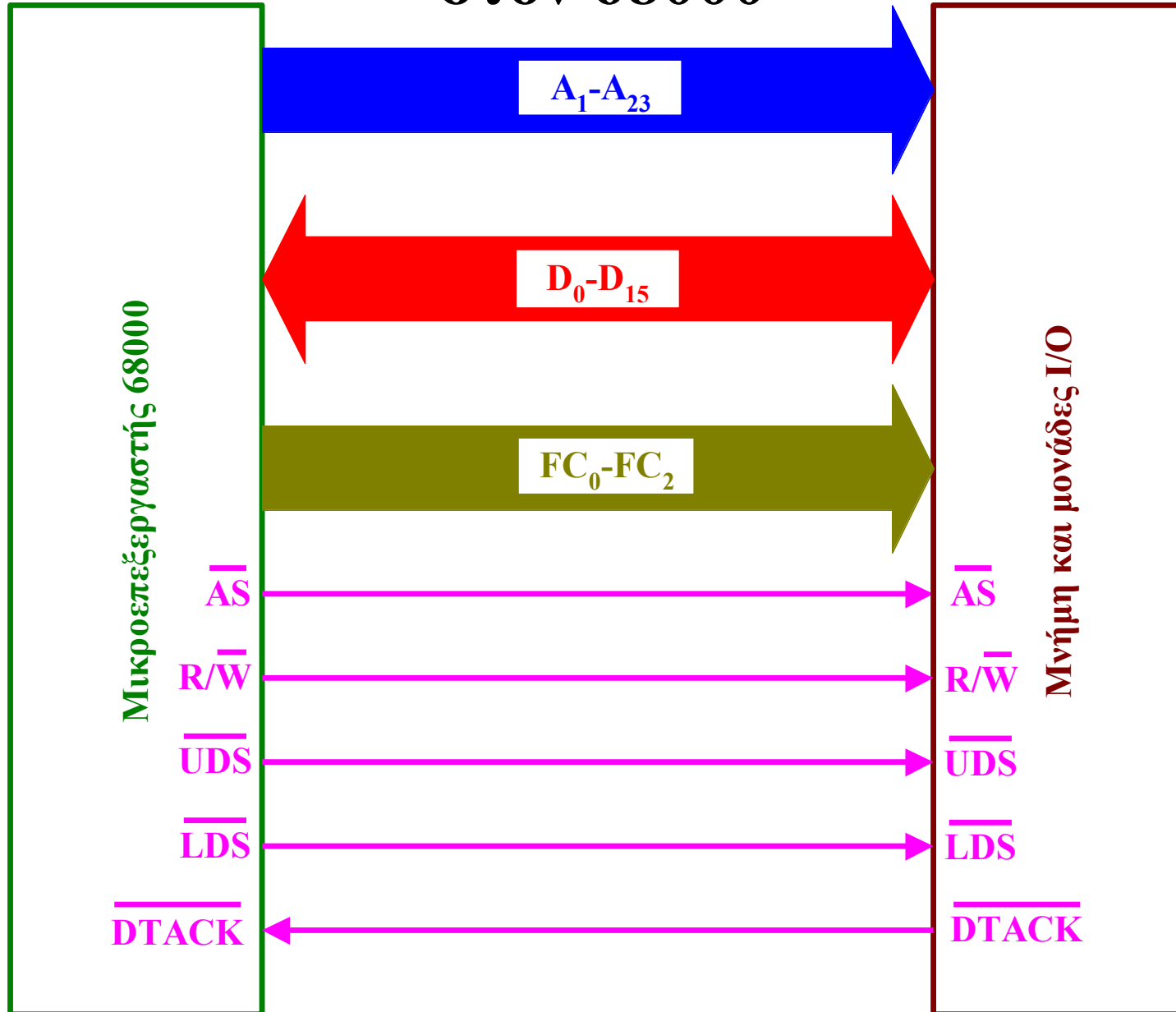
Επέκταση χωρητικότητας



Χαρτογράφηση μνήμης



Ασύγχρονη διασύνδεση μνήμης και μονάδων I/O στον 68000



Χώρος διευθύνσεων-Οργάνωση δεδομένων

Διεύθυνση
λέξης

Περιεχόμενο μνήμης

0000000₁₆

Λέξη 0

0000002₁₆

Λέξη 1

0000004₁₆

Λέξη 2

⋮

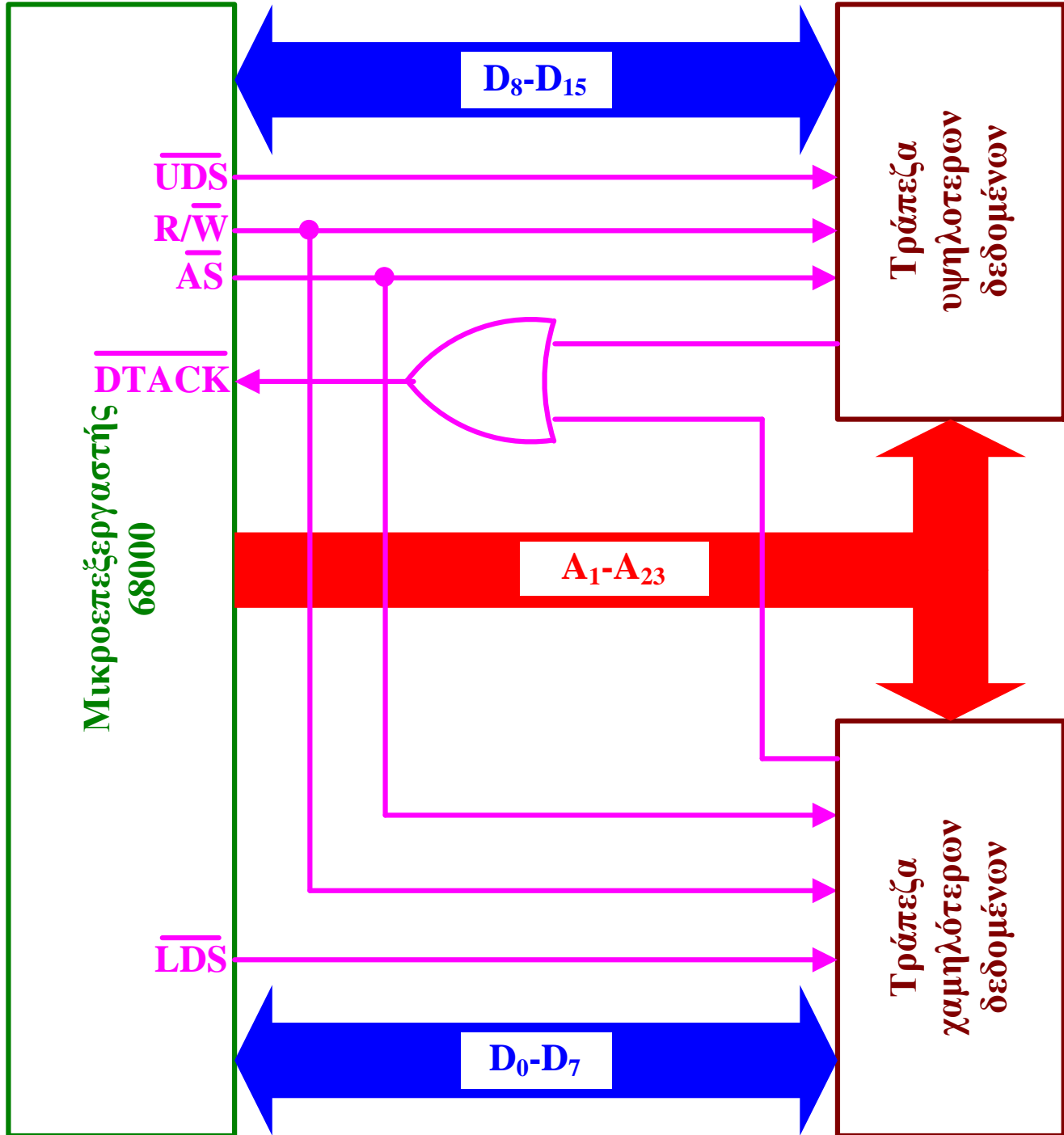
⋮

FFFFFFC₁₆

Λέξη 8.388.606

FFFFFFF₁₆

Λέξη 8.388.607



**Διεύθυνση
άρτιου byte**

Περιεχόμενο μνήμης	
Byte 0	Byte 1
Byte 2	Byte 3
Byte4	Byte5
⋮	⋮
Λέξη 8.388.604	Λέξη 8.388.605
Λέξη 8.388.606	Λέξη 8.388.607

**Διεύθυνση
περιτού byte**

0000000₁₆

0000002₁₆

0000004₁₆

⋮

FFFFFFC₁₆

FFFFFFE₁₆

0000001₁₆

0000003₁₆

0000005₁₆

⋮

FFFFFFD₁₆

FFFFFFF₁₆

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

(α)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB Byte 0 LSB								Byte 1							
Byte 2								Byte 3							

(β)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB Λέξη 0 LSB															
Λέξη 1															
Λέξη 2															

(γ)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB Μακριά Λέξη				Περισσότερο σημαντική λέξη								LSB			
0				Λιγότερο σημαντική λέξη											
MSB Μακριά Λέξη				Περισσότερο σημαντική λέξη								LSB			
1				Λιγότερο σημαντική λέξη											
MSB Μακριά Λέξη				Περισσότερο σημαντική λέξη								LSB			
2				Λιγότερο σημαντική λέξη											

(δ)

(\overline{UDS})	(\overline{LDS})	(R/\overline{W})	D_8-D_{15}	D_0-D_7
1	1	-	Άκυρα δεδομένα	Άκυρα δεδομένα
0	0	1	Έγκυρα δεδομένα d_8-d_{15}	Έγκυρα δεδομένα d_0-d_7
1	0	1	Άκυρα δεδομένα	Έγκυρα δεδομένα d_0-d_7
0	1	1	Έγκυρα δεδομένα d_8-d_{15}	Άκυρα δεδομένα
0	0	0	Έγκυρα δεδομένα d_8-d_{15}	Έγκυρα δεδομένα d_0-d_7
1	0	0	Άκυρα δεδομένα	Έγκυρα δεδομένα d_0-d_7
0	1	0	Έγκυρα δεδομένα d_8-d_{15}	Άκυρα δεδομένα

0000000₁₆
⋮
00003FE₁₆
0000400₁₆
⋮
FFFFFFF₁₆



Η μνήμη στα συστήματα που βασίζονται στο 68000 χωρίζεται σε δύο μέρη.

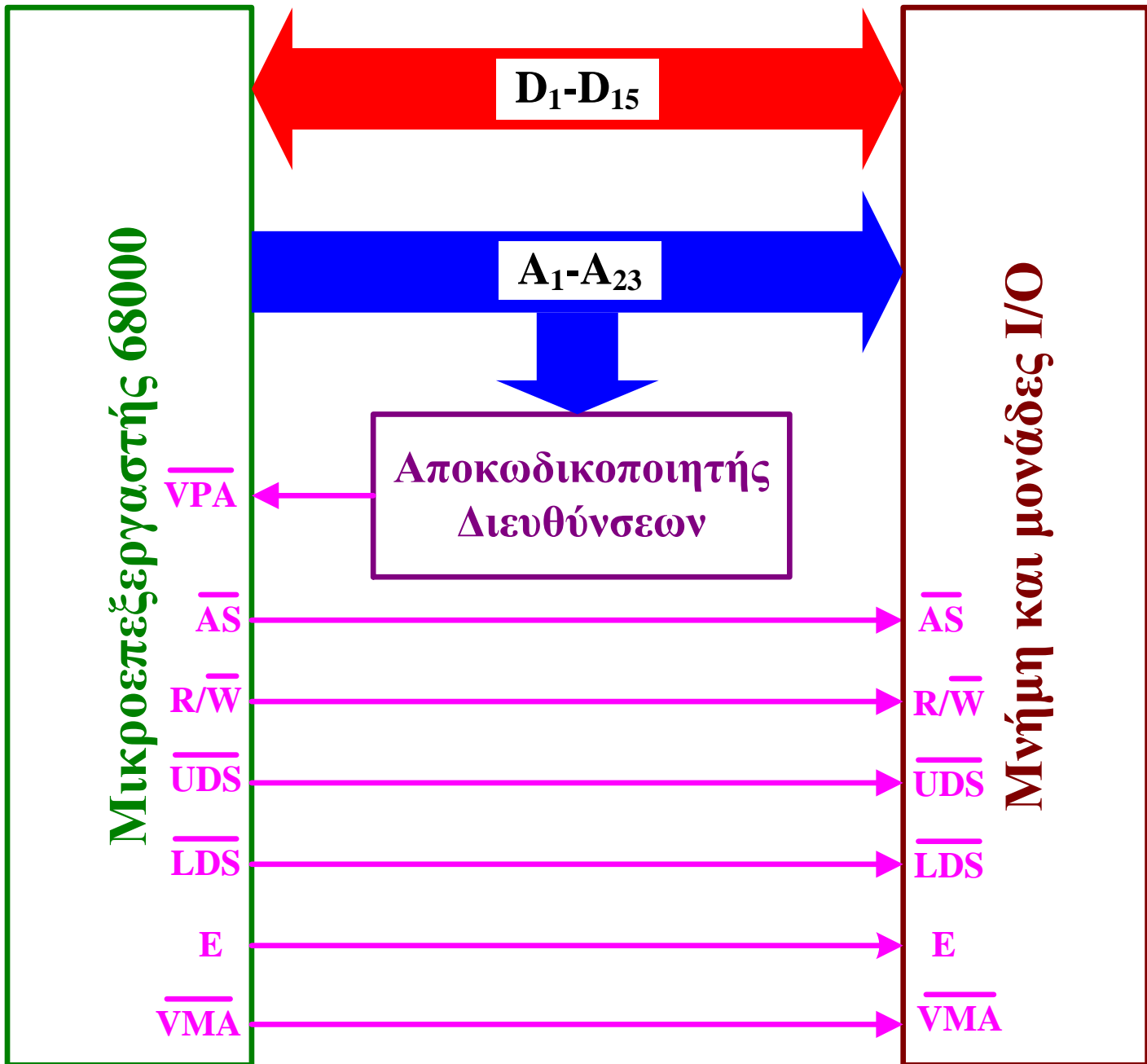
- *Στη μνήμη αποθήκευσης προγράμματος, όπου αποθηκεύεται πληροφορία όπως οι κωδικοί των εντολών του προγράμματος, άμεσες διευθύνσεις των τελεστών, και δεδομένα των τελεστών απευθείας εκκίνησης. Η μνήμη αυτή μπορεί να υλοποιηθεί με ROM ή RAM.*

- *Στη μνήμη αποθήκευσης δεδομένων, όπου αποθηκεύεται η πληροφορία των προς επεξεργασία τελεστών δεδομένων. Η μνήμη αυτή μπορεί να υλοποιηθεί μόνο με RAM, έτσι ώστε να μπορεί να γίνεται ανάγνωση των προς επεξεργασία τελεστών δεδομένων αλλά και να γίνεται εγγραφή των δεδομένων που είναι τα αποτελέσματα της επεξεργασίας αυτής.*

Το αν ο μικροεπεξεργαστής θα προσπελάσει μνήμη προγράμματος ή μνήμη δεδομένων εξαρτάται από τον κώδικα των γραμμών κατάστασης ή γραμμών λειτουργίας FC_2-FC_0 .

Κώδικας κατάστασης			Τάξη αναφοράς
FC_2	FC_1	FC_0	
0	0	0	Χωρίς αφιέρωση
0	0	1	Δεδομένα Χρήστη
0	1	0	Πρόγραμμα Χρήστη
0	1	1	Χωρίς αφιέρωση
1	0	0	Χωρίς αφιέρωση
1	0	1	Δεδομένα Επόπτη
1	1	0	Πρόγραμμα Επόπτη
1	1	1	Αναγνώριση Διακοπής

Σύγχρονη διασύνδεση μνήμης



• Στη σύγχρονη διασύνδεση χρησιμοποιούνται οι διάδρομοι και οι γραμμές ελέγχου που χρησιμοποιούνται και στη ασύγχρονη διασύνδεση με τη διαφορά ότι η γραμμή DTACK έχει αντικατασταθεί από τις σύγχρονες γραμμές ελέγχου VPA, VMA και (E).

• Η γραμμή VPA είναι είσοδος στο μικροεπεξεργαστή και όταν ενεργοποιείται (λογικό "0") σηματοδοτεί το μικροεπεξεργαστή να εκτελέσει ένα σύγχρονο κύκλο διαδρόμου. Ενεργοποιείται απ' τον αποκωδικοποιητή που φαίνεται στο κύκλωμα όταν η διεύθυνση στο διάδρομο διευθύνσεων αναφέρεται στο χώρο διευθύνσεων στις οποίες ανταποκρίνονται σύγχρονες περιφερειακές συσκευές.

• Η γραμμή VMA είναι έξοδος από το μικροεπεξεργαστή και ενεργοποιείται μόνο κατά τη διάρκεια ενός σύγχρονου κύκλου διαδρομού σηματοδοτώντας ότι στο διάδρομο υπάρχει έγκυρη πληροφορία διεύθυνσης.

• Η γραμμή E είναι στη ουσία ένα ρολόι που παράγεται εσωτερικά του μικροεπεξεργαστή και έχει συχνότητα ίση με το $1/10$ του ρολογιού του συστήματος με κύκλο τεσσάρων παλμών στην κατάσταση "1" και εξ παλμών στην κατάσταση "0". Χρησιμοποιείται στην ενεργοποίηση των εισόδων E των περιφερειακών της σειράς 6800.

Η διαδικασία που ακολουθείται όταν εκτελείται ένας σύγχρονος κύκλος διαδρομού είναι η παρακάτω:

A. Αρχικός κύκλος.

- Ο μικροεπεξεργαστής αρχίζει ένα κανονικό κύκλο ανάγνωση ή εγγραφής.

B. Ορισμός σύγχρονου κύκλου.

- Εξωτερικά κυκλώματα ενεργοποιούν τη γραμμή VPA.

Γ. Συγχρονισμός με τη γραμμή E.

- Ο μικροεπεξεργαστής ελέγχει τη γραμμή E έως ότου γίνει "0".
- Ο μικροεπεξεργαστής ενεργοποιεί τη γραμμή VMA.

Δ. Μεταφορά δεδομένων.

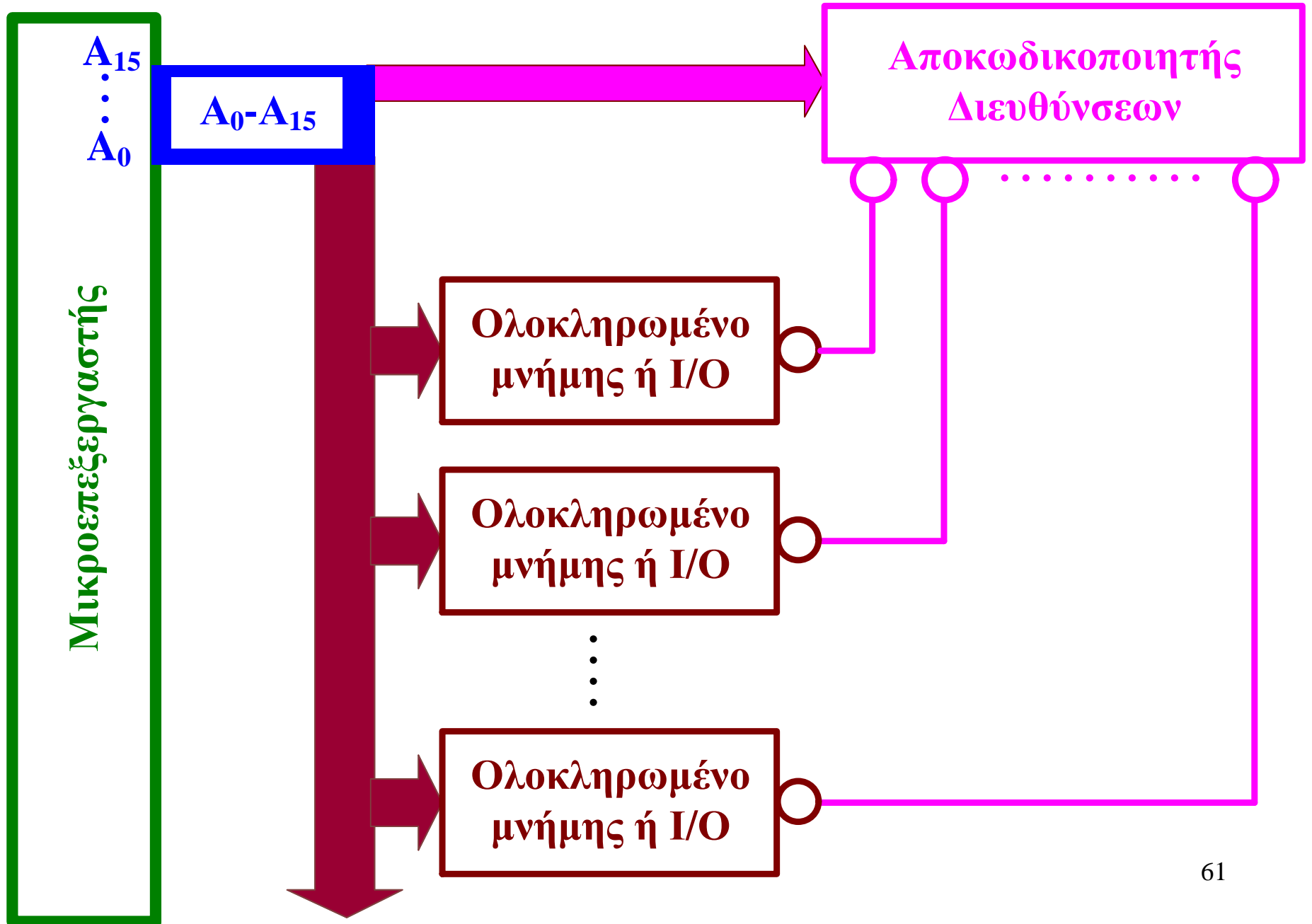
- Η περιφερειακή συσκευή περιμένει έως ότου η γραμμή E ενεργοποιηθεί και στη συνέχεια μεταφέρει τα δεδομένα.

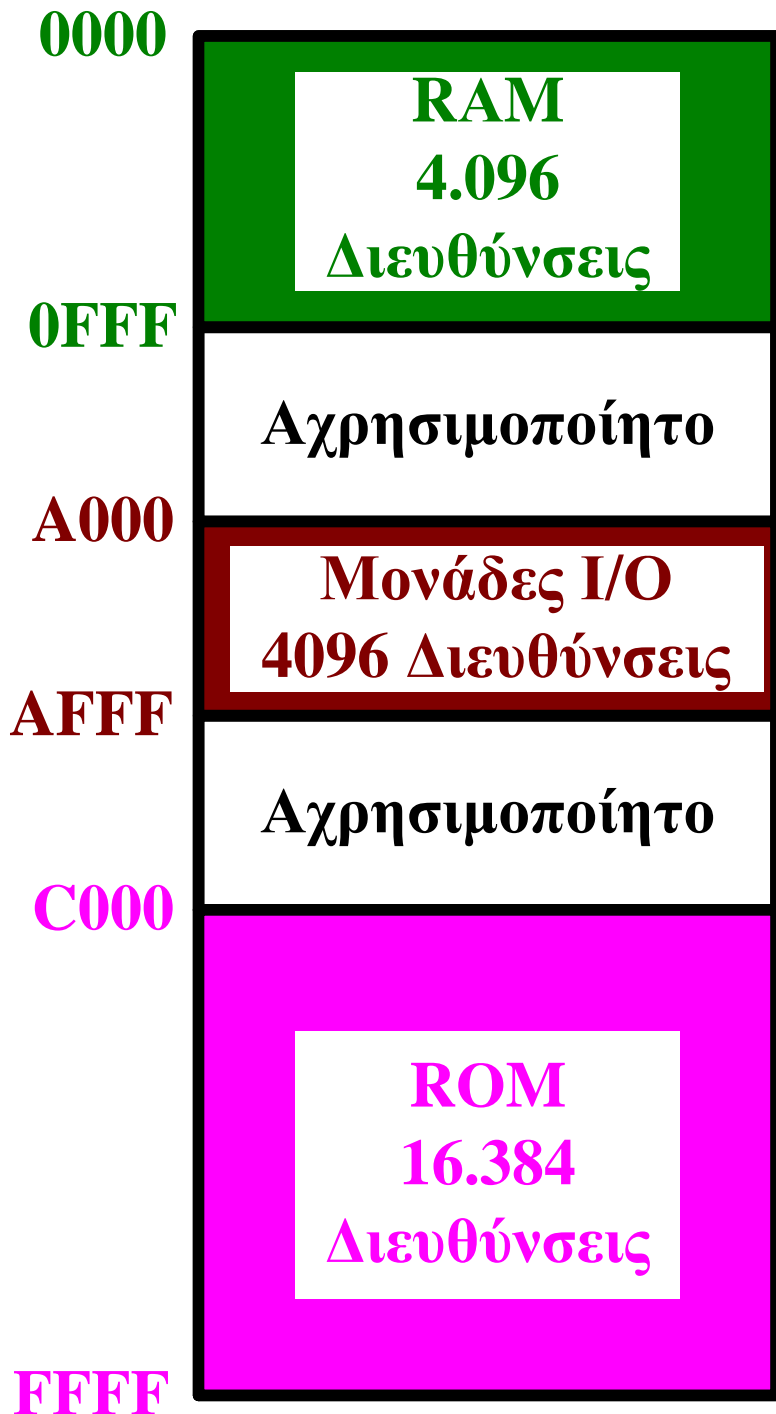
Ε. Τερματισμός κύκλου.

- Ο μικροεπεξεργαστής περιμένει έως ότου η γραμμή E γίνει "0".
- Ο μικροεπεξεργαστής απενεργοποιεί τη γραμμή VMA.
- Ο μικροεπεξεργαστής απενεργοποιεί τις γραμμές AS, UDS και LDS.

ΣΤ. Αρχίζει ο επόμενος κύκλος.

Αποκωδικοποίηση Διευθύνσεων





Πριν την διαδικασία της αποκωδικοποίησης πρέπει να γίνει η χαρτογράφηση μνήμης και να κατασκευαστεί ο πίνακας μνήμης.

Έστω ο διπλανός πίνακας μνήμης.

Αποκωδικοποίηση Μνήμης RAM

Το μέγεθος της μνήμης RAM είναι $0FFF=4096$ byte= $4k \times 8$ και αν υποθεθεί ότι επιλέγεται το ολοκληρωμένο μνήμης $1k \times 8$ (1024×8) τότε θα χρησιμοποιηθούν 4 ολοκληρωμένα και ο αριθμός των γραμμών του διαδρόμου διευθύνσεων που θα χρειαστούν για να διευθυνσιοδοτηθούν οι είσοδοι διευθύνσεων του κάθε ολοκληρωμένου ξεχωριστά και όλων μαζί είναι 10 ($2^{10}=1024$).

Άρα στις εισόδους των ολοκληρωμένων της RAM θα συνδεθούν οι γραμμές A_0-A_9 .

Οι υπόλοιπες γραμμές $A_{10}-A_{15}$ του διαδρόμου διευθύνσεων θα χρησιμοποιηθούν ως είσοδοι στο κύκλωμα αποκωδικοποίησης.

Μικροπεξεργαστής

A_{15}
⋮
 A_0

A_0-A_{15}

$A_{10}-A_{15}$

Αποκωδικοποιητής
Διευθύνσεων

A_0-A_9

Ολοκληρωμένο
RAM 1kx8

Ολοκληρωμένο
RAM 1kx8

Ολοκληρωμένο
RAM 1kx8

Ολοκληρωμένο
RAM 1kx8

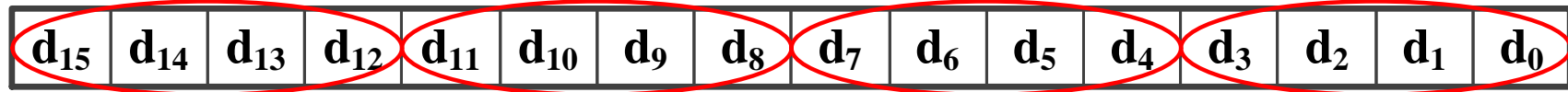
Το επόμενο βήμα θα είναι να κατασκευαστεί ο παρακάτω πίνακας.

Περιοχή διευθύνσεων	Είσοδος Γραμμές Διεύθυνσης		Έξοδος Επιλογή Ολοκληρωμένου RAM			
	A ₁₅ A ₁₄ A ₁₃ A ₁₂ A ₁₁ A ₁₀	A ₉ A ₈ A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀	<i>CS0C</i>	<i>CS08</i>	<i>CS04</i>	<i>CS00</i>
0000-03FF	L L L L L L	X X X X X X X X	H	H	H	L
0400-07FF	L L L L L H	X X X X X X X X	H	H	L	H
0800-0BFF	L L L L H L	X X X X X X X X	H	L	H	H
0C00-0FFF	L L L L H H	X X X X X X X X	L	H	H	H

[Πίσω 1](#) [Πίσω 2](#) [Πίσω 3](#) [Πίσω 4](#)

[Σχήμα](#)

Αποκωδικοποίηση διευθύνσεων



Αποκωδικοποίηση διευθύνσεων RAM

Πίνακας

d_{15}	d_{14}	d_{13}	d_{12}	d_{11}	d_{10}	d_9	d_8	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
----------	----------	----------	----------	----------	----------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

0000_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$03FF_{16}$	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
-------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Αποκωδικοποίηση διευθύνσεων RAM

Πίνακας

	d_{15}	d_{14}	d_{13}	d_{12}	d_{11}	d_{10}	d_9	d_8	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
0000_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$03FF_{16}$	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0400_{16}	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
$07FF_{16}$	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Αποκωδικοποίηση διευθύνσεων RAM

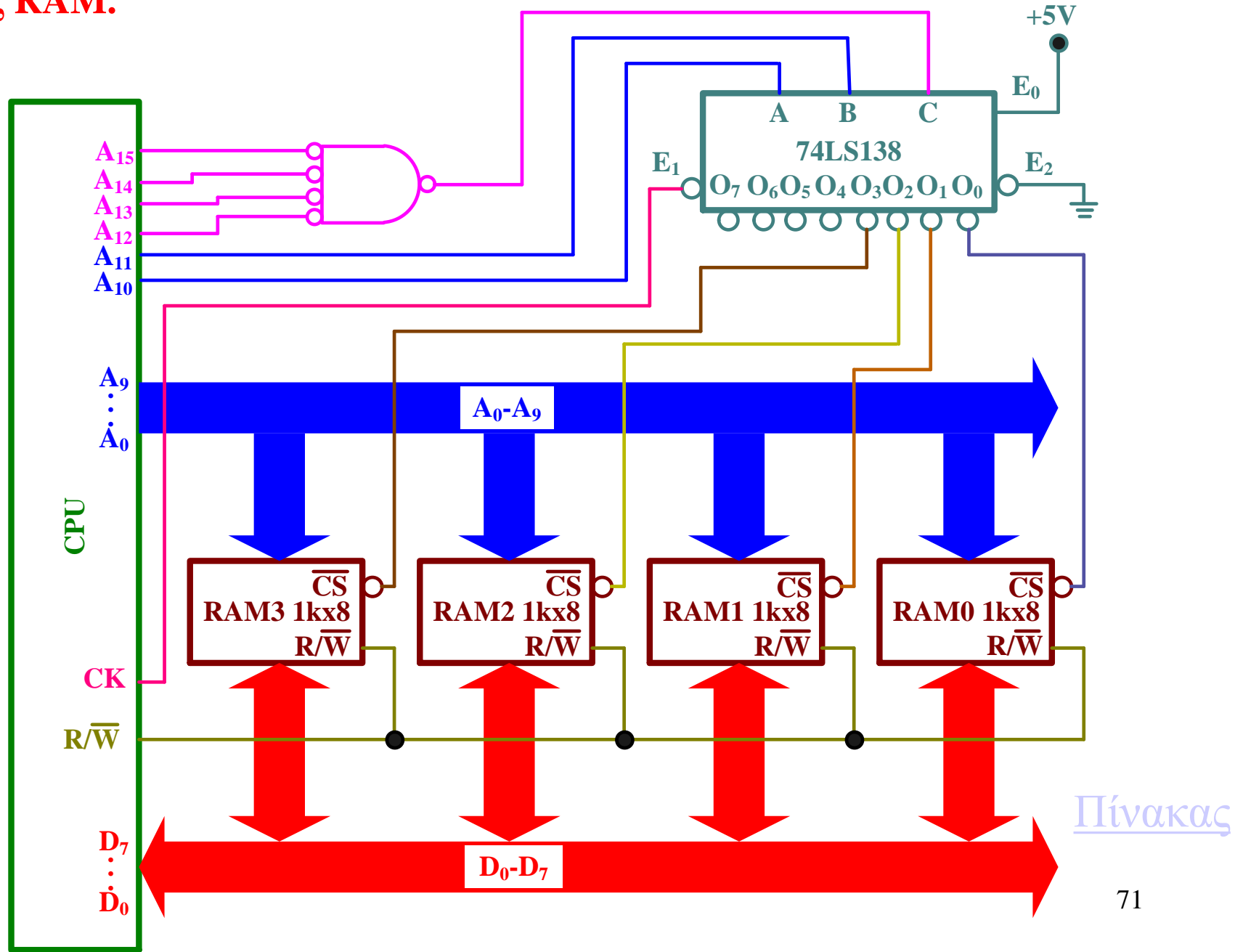
Πίνακας

	d ₁₅	d ₁₄	d ₁₃	d ₁₂	d ₁₁	d ₁₀	d ₉	d ₈	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀
0000 ₁₆	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03FF ₁₆	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0400 ₁₆	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
07FF ₁₆	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
0800 ₁₆	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0BFF ₁₆	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1

Αποκωδικοποίηση διευθύνσεων RAM

	d ₁₅	d ₁₄	d ₁₃	d ₁₂	d ₁₁	d ₁₀	d ₉	d ₈	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀
0000 ₁₆	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03FF ₁₆	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
0400 ₁₆	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
07FF ₁₆	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
0800 ₁₆	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0BFF ₁₆	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1
0C00 ₁₆	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0FFF ₁₆	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Με βάση τον πίνακα αυτό κατασκευάζεται το κύκλωμα του αποκωδικοποιητή μνήμης RAM.

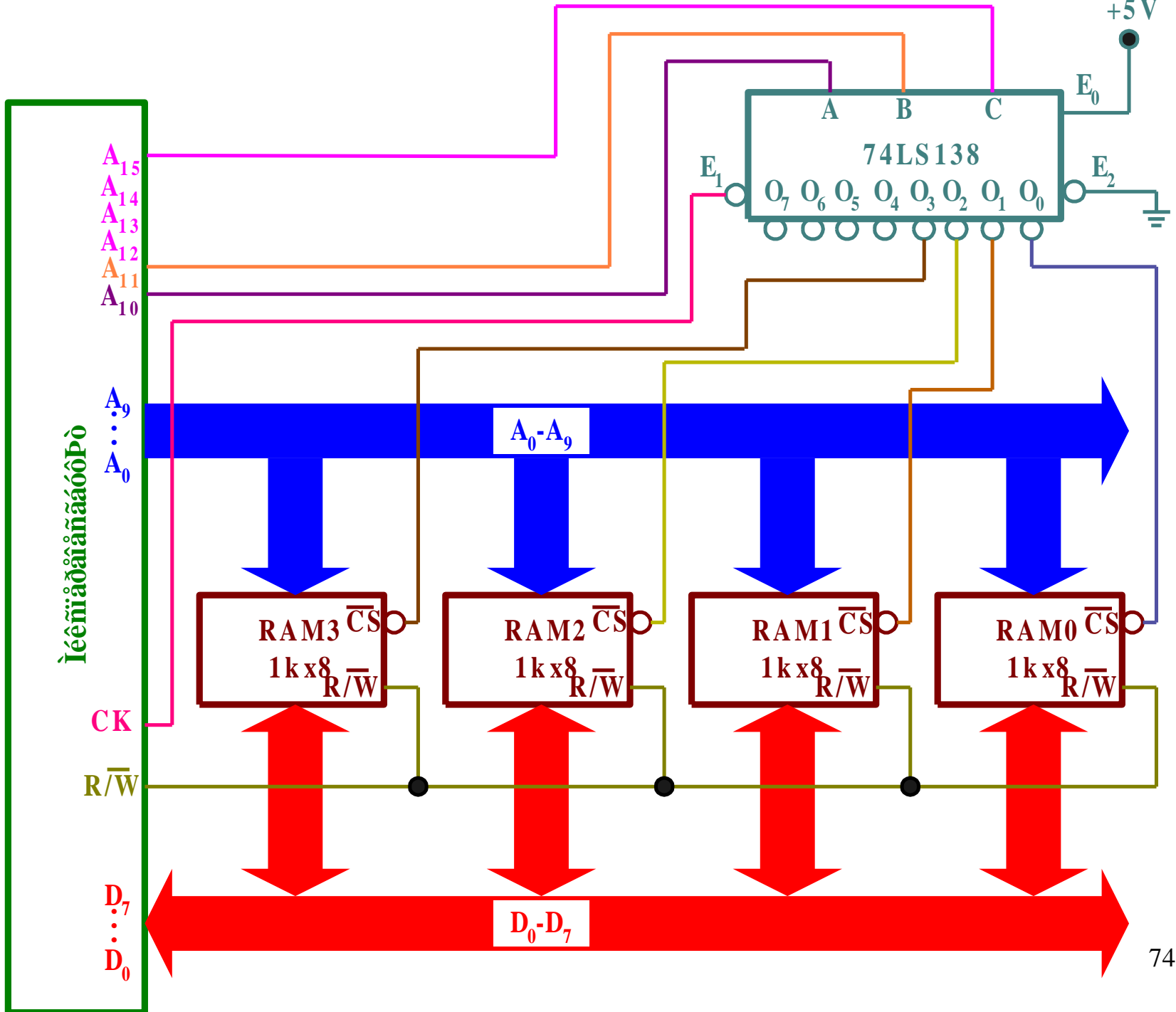


Η προηγούμενη μορφή αποκωδικοποίησης, που χρησιμοποιεί όλες τις γραμμές διευθύνσεων, λέγεται **ολική αποκωδικοποίηση (*Full Decoding*)** και χρησιμοποιείται σε συστήματα όπου όλος ο χώρος του χάρτη μνήμης χρησιμοποιείται ή πρόκειται να χρησιμοποιηθεί από το σύστημα.

Στις περισσότερες όμως εφαρμογές, μέρος μόνο του χάρτη μνήμης χρησιμοποιείται.

Στις περιπτώσεις αυτές χρησιμοποιείται η μέθοδος της μερικής αποκωδικοποίησης (*Partial Decoding*), που φαίνεται στο επόμενο κύκλωμα, όπου η πύλη OR έχει απαλειφθεί και έχει αντικατασταθεί με απευθείας σύνδεση της γραμμής A_{15} στον αποκωδικοποιητή.

Οι γραμμές $A_{14}A_{13}A_{12}$ δεν χρησιμοποιούνται και επομένως δεν επηρεάζουν τη λειτουργία του αποκωδικοποιητή.



ÍéñĩãðãĩãããóõPò

A_{15}
 A_{14}
 A_{13}
 A_{12}
 A_{11}
 A_{10}
 A_9
 \vdots
 A_0

 CK
 R/\overline{W}

 D_7
 \vdots
 D_0

A_0-A_9

D_0-D_7

RAM3 \overline{CS}
1kx8
 R/\overline{W}

RAM2 \overline{CS}
1kx8
 R/\overline{W}

RAM1 \overline{CS}
1kx8
 R/\overline{W}

RAM0 \overline{CS}
1kx8
 R/\overline{W}

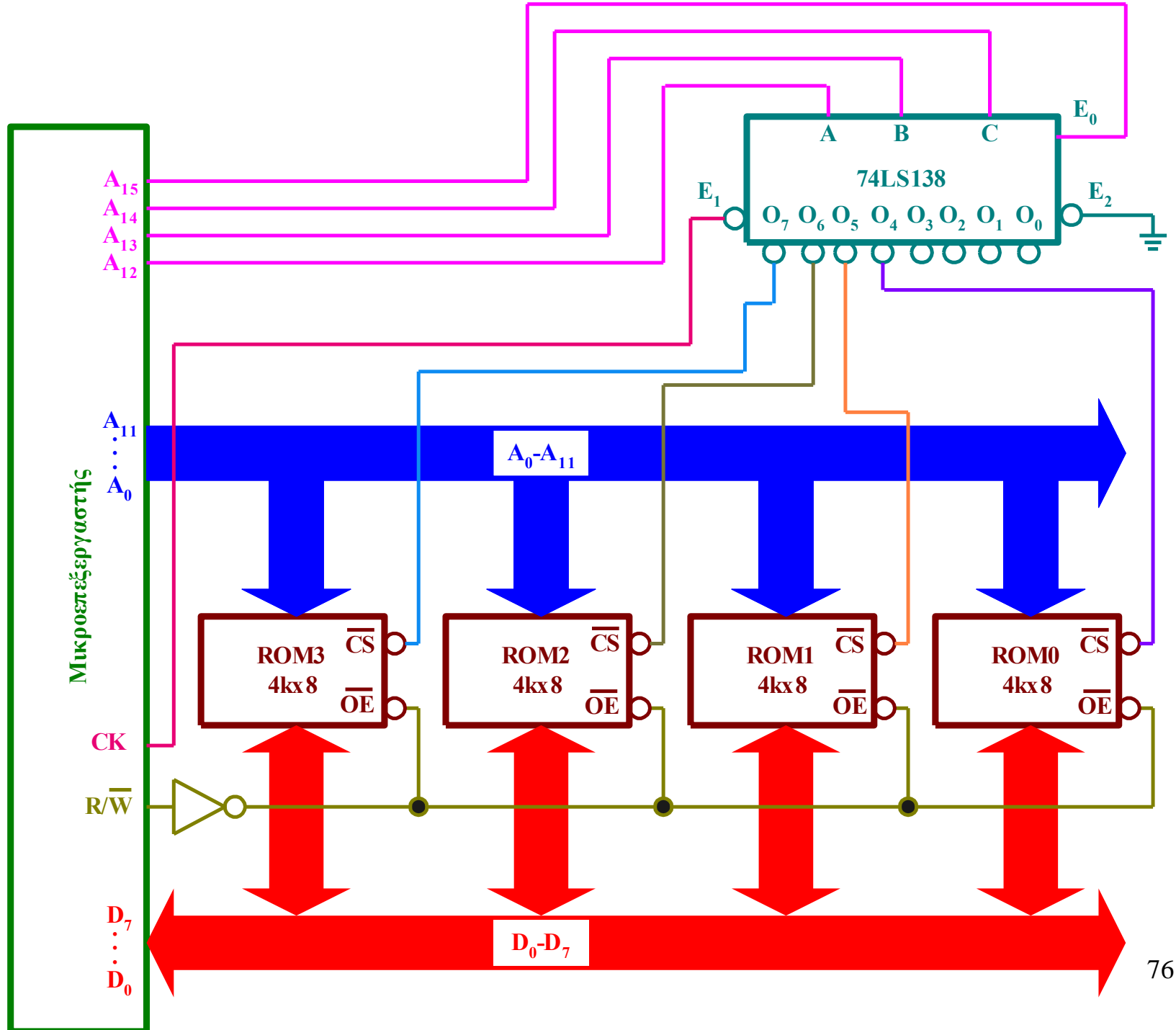
74LS138

Αποκωδικοποίηση Μνήμης ROM

Το μέγεθος της μνήμης ROM είναι (4FFF) 16294 byte 16kx8 και αν υποθεθεί ότι επιλέγεται το ολοκληρωμένο μνήμης 4kx8 τότε ο αριθμός των γραμμών του διαδρόμου διευθύνσεων που θα χρειαστούν για να διευθυνσιοδοτηθούν οι είσοδοι διευθύνσεων του είναι 12 ($2^{12}=4096$).

Το επόμενο βήμα θα είναι να κατασκευαστεί ο παρακάτω πίνακας.

Περιοχή διευθύνσεων	Είσοδος Γραμμές Διεύθυνσης										Έξοδος Επιλογή Ολοκληρωμένου ROM			
	A ₁₅ A ₁₄ A ₁₃ A ₁₂	A ₁₁ A ₁₀ A ₉ A ₈ A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀	C _{SF}	C _{SE}	C _{SD}	C _{SC}								
C000-CFFF	H H L L	X X X X X X X X X X X X	H	H	H	L								
D000-DFFF	H H L H	X X X X X X X X X X X X	H	H	L	H								
E000-EFFF	H H H L	X X X X X X X X X X X X	H	L	H	H								
F000-FFFF	H H H H	X X X X X X X X X X X X	L	H	H	H								

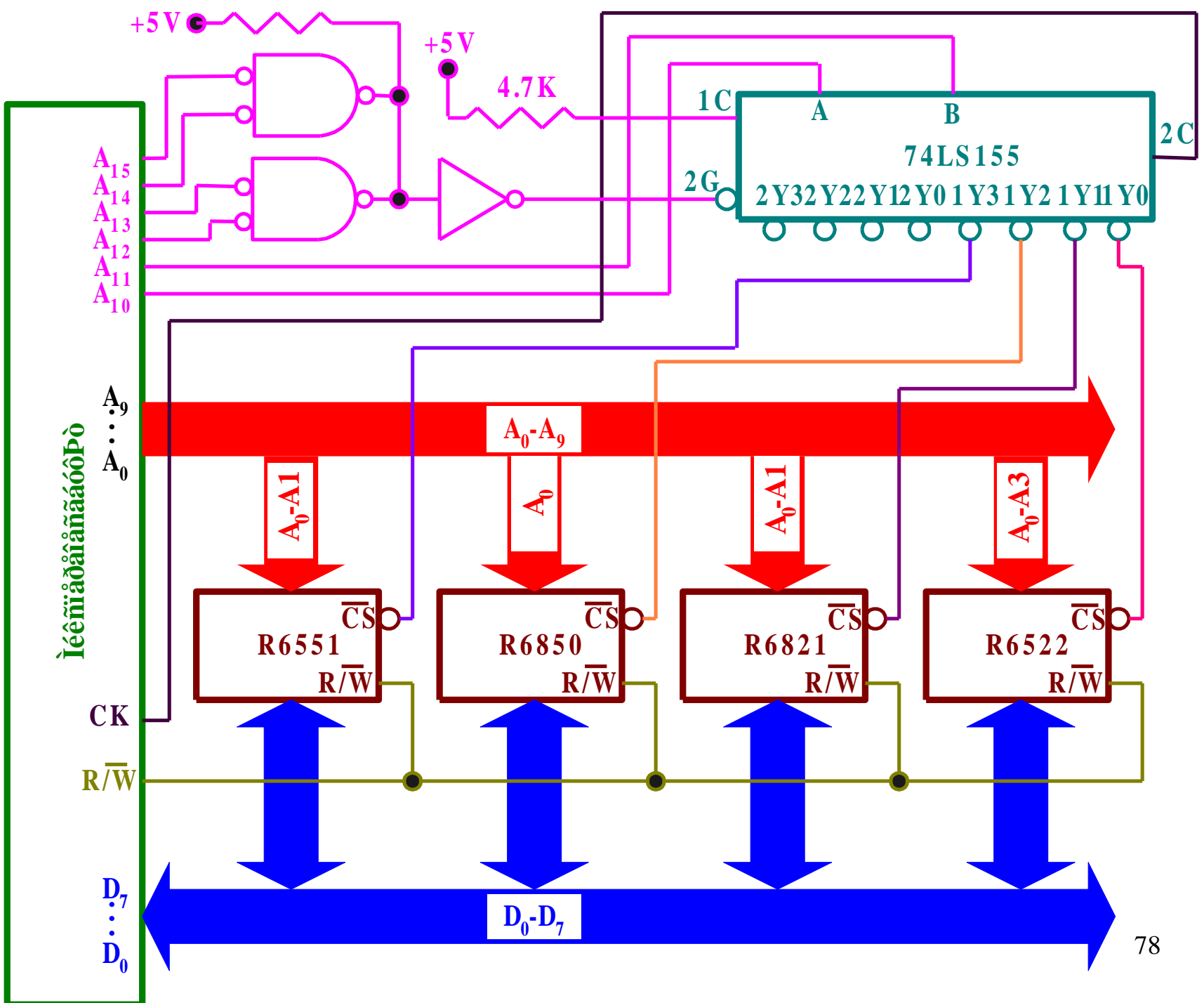


Αποκωδικοποίηση μονάδων I/O

Αν υποθεθεί και πάλι ο ίδιος μικροεπεξεργαστής και ότι τα ολοκληρωμένα εισόδου-εξόδου που πρόκειται να χρησιμοποιηθούν είναι τα ολοκληρωμένα παράλληλης προσαρμογής R6522 και R6821 καθώς και τα ολοκληρωμένα σειριακής προσαρμογής R6850 και R6551.

Το επόμενο βήμα θα είναι να κατασκευαστεί ο παρακάτω πίνακας.

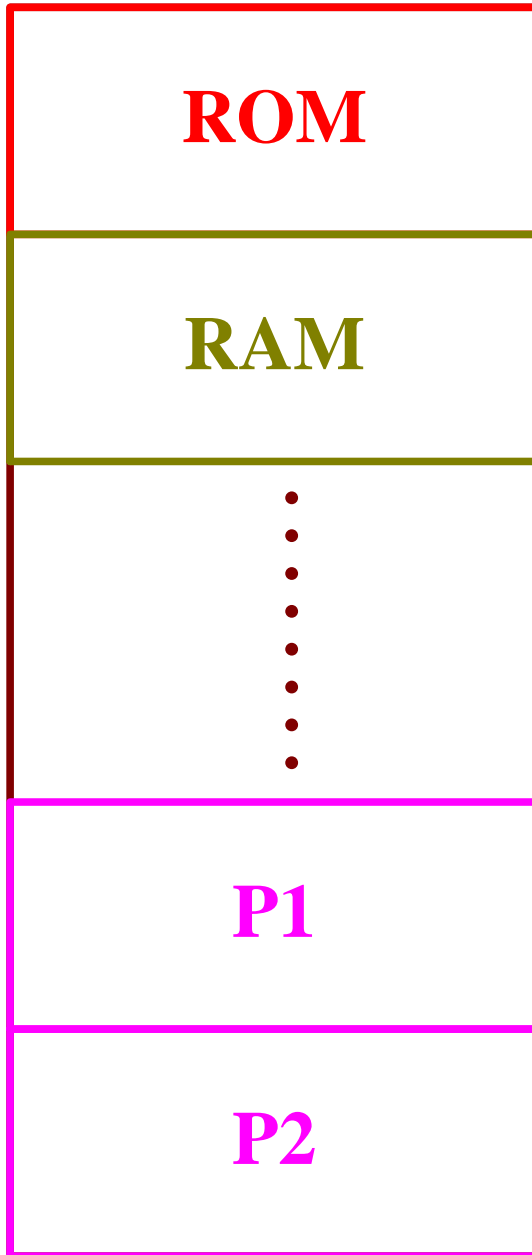
Περιοχή διευθύνσεων	Είσοδος								Έξοδος			
	Γραμμές Διεύθυνσης								Επιλογή Ολοκληρωμένων I/O			
	A ₁₅ A ₁₄ A ₁₃ A ₁₂	A ₁₁ A ₁₀ A ₉ A ₈ A ₇ A ₆ A ₅ A ₄ A ₃ A ₂ A ₁ A ₀	CSAC	CSA8	CSA4	CSA0						
A000-A3FF	H L H L	L L X X X X X X X X	H	H	H	L						
A400-A7FF	H L H L	L H X X X X X X X X	H	H	L	H						
A800-ABFF	H L H L	H L X X X X X X X X	H	L	H	H						
AC00-AFFF	H L H L	H H X X X X X X X X	L	H	H	H						



Σχεδίαση αποκωδικοποιητή μνήμης και μονάδων εισόδου-εξόδου με PLD

Ας υποθεθεί ότι χρειάζεται να σχεδιαστεί ένα κύκλωμα αποκωδικοποίησης διευθύνσεων που θα αποκωδικοποιήσει το χάρτη μνήμης, ενός συστήματος βασισμένου στο μικροεπεξεργαστή 68000.

0000000



16kbytes (8kwords) ROM

003FFF

ROM

004000

RAM

16kbytes (8kwords) RAM

007FFF

•
•
•
•
•
•
•
•

•
•
•
•
•
•
•
•

P1

Δύο περιφερειακές συσκευές που χρειάζονται τουλάχιστο 16 λέξεις (32 bytes) η κάθε μια και χρησιμοποιούν μόνο τις περιττές διευθύνσεις

P2

FFFFFFF

Από τον πίνακα μνήμης κατασκευάζεται ο πίνακας που δείχνει τις τιμές που πρέπει να πάρουν οι γραμμές του διαδρόμου διευθύνσεων για να υλοποιηθεί το κύκλωμα του αποκωδικοποιητή.

A ₂₃	A ₂₂	A ₂₁	A ₂₀	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	000000	ROM
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	003FFF	ROM
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	004000	RAM
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	007FFF	RAM
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	008000	Κενό
1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFDFFF	Κενό
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FFE000	PER1
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	FFEFFF	PER1
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	FFF000	PER2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFF	PER2



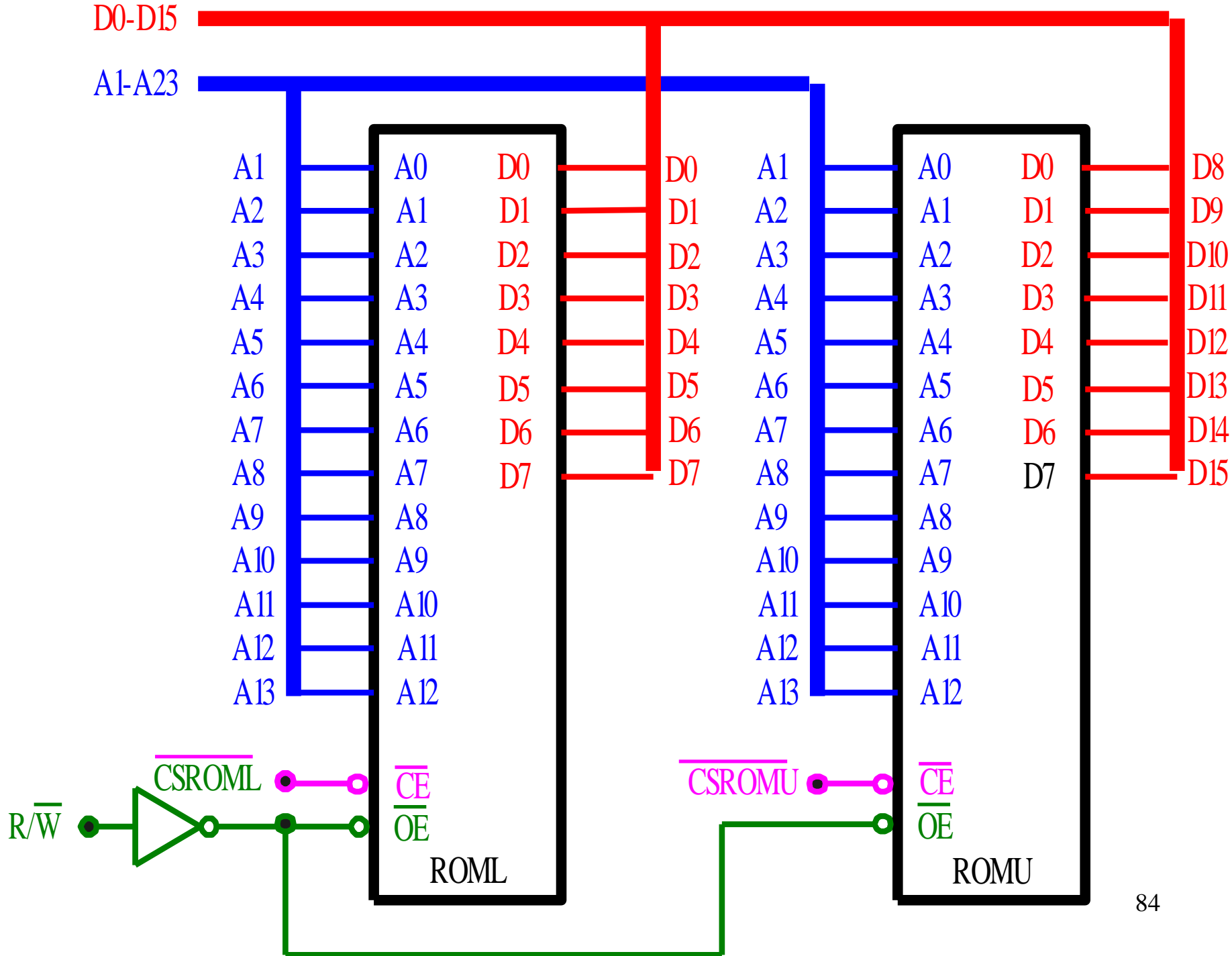
Οι γραμμές του μικροεπεξεργαστή που πρέπει να χρησιμοποιηθούν ως είσοδοι στην FPLA ή την PAL είναι:

- Οι γραμμές UDS και LDS, που θα επιλέξουν αντίστοιχα τα υψηλά και χαμηλά bytes της μνήμης καθώς και τις περιττές διευθύνσεις (μόνο η γραμμή LDS) των περιφερειακών μονάδων.
- Η γραμμή R/W, που θα βεβαιώνει ότι η ROM επιλέγεται μόνο κατά τη διάρκεια ενός κύκλου ανάγνωσης.
- Η γραμμή VMA, που θα επιλέξει τις περιφερειακές συσκευές.

Σπόνδυλος ROM

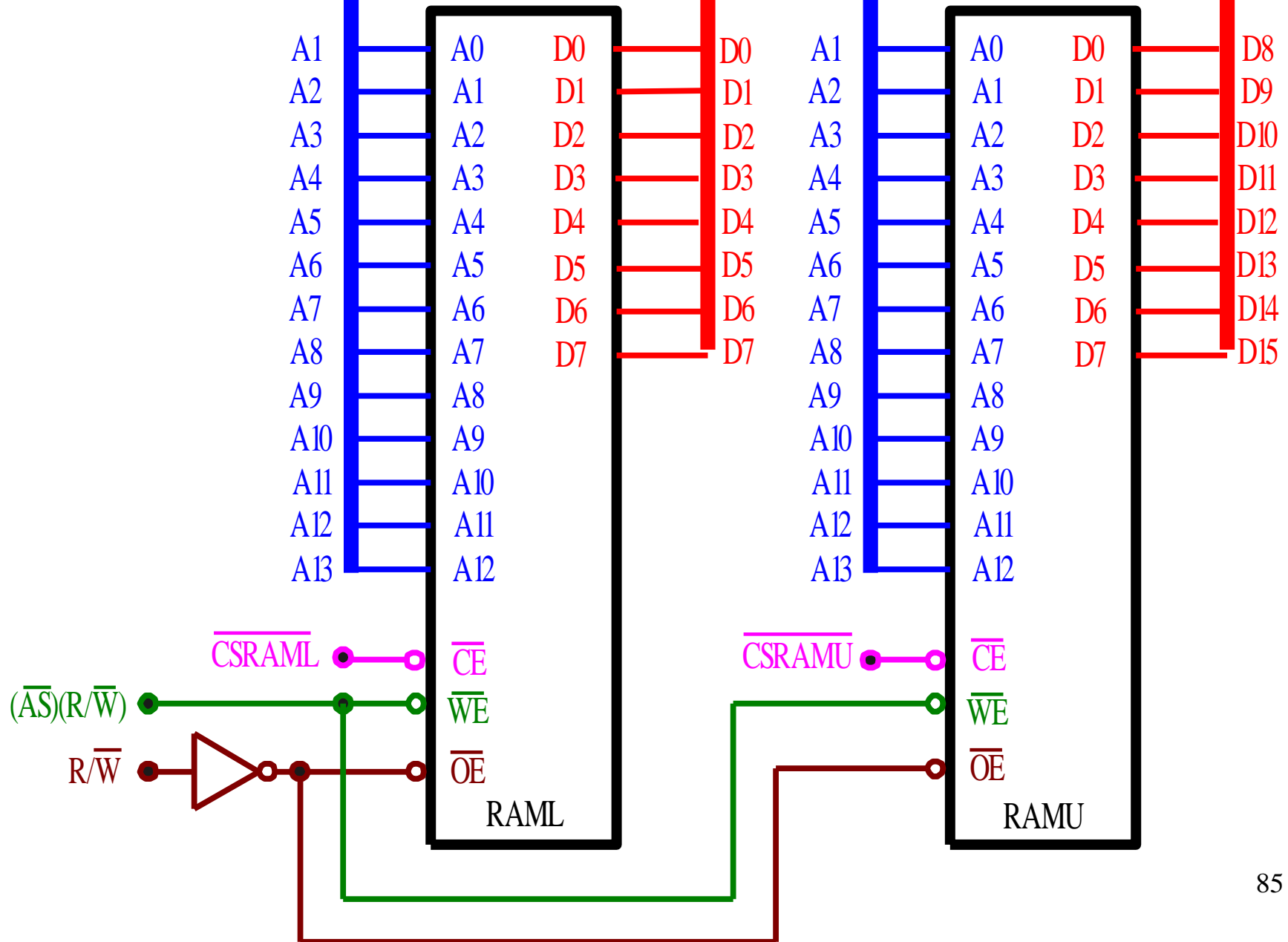
Για την αποκωδικοποίηση της ROM:

- Επιλέγονται δύο ολοκληρωμένα μνήμης 8kx8 ROM που θα συνδεθούν ώστε να αποτελέσουν σπόνδυλο μνήμης 8kx16.
- Στο σπόνδυλο μνήμης ROM διατίθενται οι διευθύνσεις 000000_{16} - $003FFF_{16}$
- Οι γραμμές του διαδρόμου διευθύνσεων που θα χρειαστούν να διευθυνσιοδοτήσουν 8k λέξεις μνήμης είναι 13 από A_1 έως A_{13} , που όταν θα συνδυαστούν με τις γραμμές LDS και UDS (ψεύδο A_0) θα διευθυνσιοδοτήσουν 16K byte μνήμης.



D0-D15

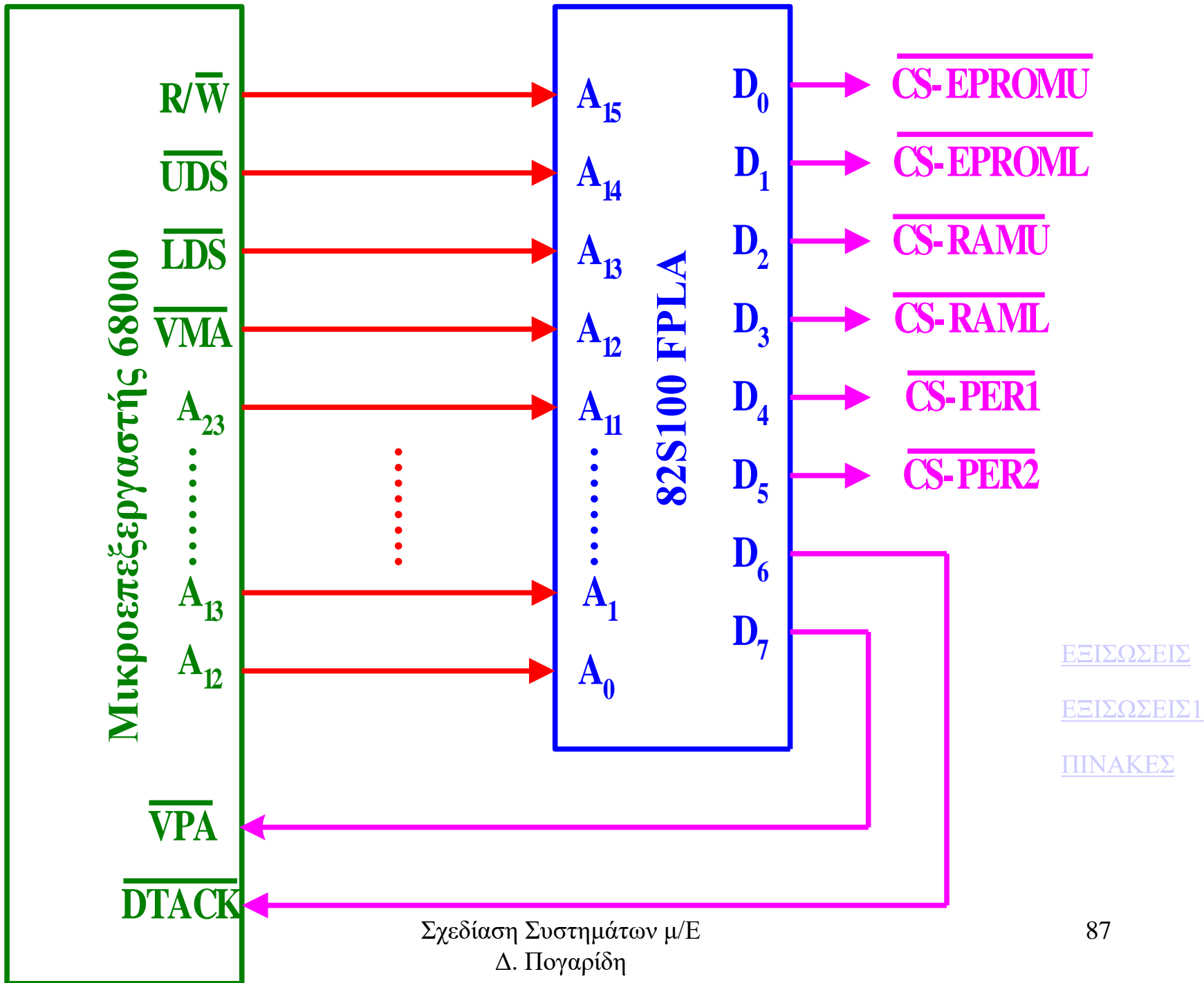
A1-A23



	A ₂₃	A ₂₂	A ₂₁	A ₂₀	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁			
0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	000000	EPROM
0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	003FFF	
0	0	0	0	0	0	0	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	004000	RAM
0	0	0	0	0	0	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	007FFF	
1	1	1	1	1	1	1	1	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	FFE000	PER1
1	1	1	1	1	1	1	1	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	FFEFFF	
1	1	1	1	1	1	1	1	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	FFF000	PER2
1	1	1	1	1	1	1	1	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	FFFFFF	

Η γραμμή DTACK ενεργοποιείται κάθε φορά που στις γραμμές του διαδρόμου εμφανίζεται μία διεύθυνση μεταξύ 000000₁₆ και 007FFF₁₆.

R/W	UDS	LDS	VMA	A ₂₃	A ₂₂	A ₂₁	A ₂₀	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	A ₁₃	A ₁₂	68000	FPLA
A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	FPLA	ΕΞΙΣΩΣΕΙΣ
1	0	X	X	0	0	0	0	0	0	0	0	0	0	X	X	EPROMU	ΕΞΙΣΩΣΕΙΣ1
1	X	0	X	0	0	0	0	0	0	0	0	0	0	X	X	EPROML	
X	0	X	X	0	0	0	0	0	0	0	0	0	1	X	X	RAMU	
X	X	0	X	0	0	0	0	0	0	0	0	0	1	X	X	RAML	
X	X	X	X	0	0	0	0	0	0	0	0	0	X	X	X	DTACK	
X	X	0	0	1	1	1	1	1	1	1	1	1	1	1	0	PER1	
X	X	0	0	1	1	1	1	1	1	1	1	1	1	1	1	PER2	
X	X	0	X	1	1	1	1	1	1	1	1	1	1	X	X	VPA	



Η FPLA πρέπει να προγραμματιστεί ώστε στις εξόδους της παράταξης των πυλών AND να παρθούν οι όροι γινομένων:

$$P0 = \overline{R/W} \cdot \overline{UDS} \cdot \overline{A_{14}} \cdot \overline{A_{15}} \cdot \overline{A_{16}} \cdot \overline{A_{17}} \cdot \overline{A_{18}} \cdot \overline{A_{19}} \cdot \overline{A_{20}} \cdot \overline{A_{21}} \cdot \overline{A_{22}} \cdot \overline{A_{23}}$$

$$P1 = \overline{R/W} \cdot \overline{LDS} \cdot \overline{A_{14}} \cdot \overline{A_{15}} \cdot \overline{A_{16}} \cdot \overline{A_{17}} \cdot \overline{A_{18}} \cdot \overline{A_{19}} \cdot \overline{A_{20}} \cdot \overline{A_{21}} \cdot \overline{A_{22}} \cdot \overline{A_{23}}$$

$$P2 = \overline{UDS} \cdot \overline{A_{14}} \cdot \overline{A_{15}} \cdot \overline{A_{16}} \cdot \overline{A_{17}} \cdot \overline{A_{18}} \cdot \overline{A_{19}} \cdot \overline{A_{20}} \cdot \overline{A_{21}} \cdot \overline{A_{22}} \cdot \overline{A_{23}}$$

$$P3 = \overline{LDS} \cdot \overline{A_{14}} \cdot \overline{A_{15}} \cdot \overline{A_{16}} \cdot \overline{A_{17}} \cdot \overline{A_{18}} \cdot \overline{A_{19}} \cdot \overline{A_{20}} \cdot \overline{A_{21}} \cdot \overline{A_{22}} \cdot \overline{A_{23}}$$

$$P4 = \overline{LDS} \cdot \overline{VMA} \cdot \overline{A_{12}} \cdot \overline{A_{13}} \cdot \overline{A_{14}} \cdot \overline{A_{15}} \cdot \overline{A_{16}} \cdot \overline{A_{17}} \cdot \overline{A_{18}} \cdot \overline{A_{19}} \cdot \overline{A_{20}} \cdot \overline{A_{21}} \cdot \overline{A_{22}} \cdot \overline{A_{23}}$$

$$P5 = \overline{LDS} \cdot \overline{VMA} \cdot \overline{A_{12}} \cdot \overline{A_{13}} \cdot \overline{A_{14}} \cdot \overline{A_{15}} \cdot \overline{A_{16}} \cdot \overline{A_{17}} \cdot \overline{A_{18}} \cdot \overline{A_{19}} \cdot \overline{A_{20}} \cdot \overline{A_{21}} \cdot \overline{A_{22}} \cdot \overline{A_{23}}$$

$$P6 = \overline{LDS} \cdot \overline{A_{14}} \cdot \overline{A_{15}} \cdot \overline{A_{16}} \cdot \overline{A_{17}} \cdot \overline{A_{18}} \cdot \overline{A_{19}} \cdot \overline{A_{20}} \cdot \overline{A_{21}} \cdot \overline{A_{22}} \cdot \overline{A_{23}}$$

όπου, $P0=EPROMU$, $P1=EPROML$, $P2=RAMU$, $P3=RAML$,
 $P4=PER1$, $P5=PER2$, και $P6=VPA$

[ΠΙΝΑΚΕΣ](#)

[FPLA](#)

[ΕΞΙΣΩΣΕΙΣ1](#)

Με βάση τα προηγούμενα, η παράταξη των πυλών OR πρέπει να προγραμματιστεί έτσι ώστε:

$$CSEPR0MU: D0 = \overline{P0}$$

$$CSEPR0ML: D1 = \overline{P1}$$

$$CSR0AMU: D2 = \overline{P2}$$

$$CSR0AML: D3 = \overline{P3}$$

$$CSPER1: D4 = \overline{P4}$$

$$CSPER2: D5 = \overline{P5}$$

$$\overline{DTACK}: D6 = \overline{P0 + P1 + P2 + P3}$$

$$\overline{VPA}: D7 = \overline{P6}$$

[FPLA](#)

[ΕΞΙΣΩΣΕΙΣ](#)

[ΠΙΝΑΚΕΣ](#)

Ερωτήσεις Θεωρίας για τα Συστήματα Μνήμης

Δύο ολοκληρωμένα EPROM χρησιμοποιούνται για το σχηματισμό μιας μνήμης 4kWords.

α. Πόσες γραμμές διευθύνσεων θα χρησιμοποιηθούν απ' τις EPROM;

12 Γραμμές (A1-A12)

β. Ποιες γραμμές διευθύνσεων θα χρησιμοποιηθούν απ' τον αποκωδικοποιητή;

A13-A23

Δύο ολοκληρωμένα EPROM χρησιμοποιούνται για το σχηματισμό μιας μνήμης 8kWords.

α. Πόσες γραμμές διευθύνσεων θα χρησιμοποιηθούν απ' τις EPROM;

13 Γραμμές (A1-A12)

β. Ποιες γραμμές διευθύνσεων θα χρησιμοποιηθούν απ' τον αποκωδικοποιητή;

A14-A23

Μια μνήμη RAM αποτελείται από δύο ολοκληρωμένα 32kx8 συνδεδεμένα ώστε να σχηματίζουν μια μνήμη 32k ψηφιολέξεων των 16 ψηφίων.

- α. Πόσες γραμμές εισόδου-εξόδου δεδομένων έχει; 16 Γραμμές
- β. Πόσες εισόδους διευθύνσεων χρειάζεται; 15 εισόδους διευθύνσεων
- γ. Ποια είναι η χωρητικότητα αυτή της μνήμης σε byte; 64Kbyte

Μια μνήμη μπορεί να αποθηκεύσει 16k ψηφιολέξεις των 16 ψηφίων.

- α. Πόσες γραμμές εισόδου-εξόδου δεδομένων έχει; 16 γραμμές
- β. Πόσες εισόδους διευθύνσεων χρειάζεται; 14 γραμμές
- γ. Ποια είναι η χωρητικότητα αυτή της μνήμης σε byte; 32k

Ποια από τα παρακάτω είναι αληθινά και ποια όχι:

- α. Η μνήμη MROM μπορεί να προγραμματιστεί από τον κατασκευαστή. Σωστό
- β. Η μνήμη PROM μπορεί να σβηστεί με το να τοποθετηθεί σε υπεριώδη ακτινοβολία για χρόνο 20 sec και να επαναπρογραμματιστεί. Λάθος
- γ. Μέρος μόνο της μνήμης EPROM μπορεί να σβηστεί και να επαναπρογραμματιστεί. Λάθος
- δ. Η μνήμη EAROM χρειάζεται να βγει από το κύκλωμα για να επαναπρογραμματιστεί. Λάθος
- ε. Μέρος μόνο της μνήμης EAROM μπορεί να σβηστεί και να επαναπρογραμματιστεί. Σωστό