

ΔΗΜΗΤΡΗΣ ΝΙΚΟΛΟΣ

Καθηγητής Πολυτεχνικής Σχολής Πανεπιστημίου Πατρών

**ΑΣΚΗΣΕΙΣ
ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ
ΥΠΟΛΟΓΙΣΤΩΝ**

ΑΣΚΗΣΕΙΣ

ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Δημήτρης Νικολός, Καθηγητής
Τμήμα Μηχανικών Ηλεκτρονικών
Υπολογιστών και Πληροφορικής
Πολυτεχνική Σχολή Πανεπιστημίου Πατρών



Γκιούρδας
Εκδοτική

ΑΣΚΗΣΕΙΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Συγγραφέας: Δημήτριος Νικολός

Υπεύθυνη Παραγωγής: Φωτεινή Σασσάλου

Σελιδοποίηση: Γεώργιος Γοζαδίνος

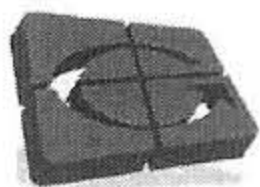
Σχεδιασμός Εξωφύλλου: Αναστασίου Χαρά

© Copyright 2010, Γκιούρδας Εκδοτική

Στοιχεία Εκτύπωσης:

1^η Έκδοση, Ιανουάριος 2010

ISBN 13: 978-960-387-902-2



**Γκιούρδας
Εκδοτική**

Δήμητρας 9, 10442 Αθήνα

τηλ.: 0030 210 5158010, fax: 0030 210 5158011

www.giourdas.com, e-mail: info@giourdas.com

Απαγορεύεται η με οποιονδήποτε τρόπο, μέσο και μέθοδο αναδημοσίευση, αναπαραγωγή, μετάφραση, διασκευή, θέση σε κυκλοφορία και η εν γένει πάσης φύσεως χρήση και εκμετάλλευση του παρόντος έργου στο σύνολό του ή τμηματικά σύμφωνα με τις διατάξεις της υπάρχουσας νομοθεσίας περί προστασίας πνευματικής ιδιοκτησίας.

Για την παραγωγή και την εκτύπωση του παρόντος συγγράμματος κατεβλήθη κάθε δυνατή προσπάθεια δημιουργίας ενός άρτιου προϊόντος. Ως εκ τούτου, ο εκδότης και ο συγγραφέας ουδεμία ευθύνη φέρουν για τυχόν λάθη ή παραλείψεις του περιεχομένου του.

*Αφιερώνεται
στην οικογένειά μου*

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1

Ασκήσεις στη δομή, οργάνωση, λειτουργία και αξιολόγηση υπολογιστών 1

ΚΕΦΑΛΑΙΟ 2

Ασκήσεις στην οργάνωση και διαχείριση της πληροφορίας στον υπολογιστή 13

ΚΕΦΑΛΑΙΟ 3

Ασκήσεις για την Κεντρική Μονάδα Επεξεργασίας 47

ΚΕΦΑΛΑΙΟ 4

Ασκήσεις για επεξεργαστές μερικώς επικαλυπτόμενων λειτουργιών 119

ΚΕΦΑΛΑΙΟ 5

Ασκήσεις για το σύστημα μνήμης 157

ΚΕΦΑΛΑΙΟ 6

Ασκήσεις για το σύστημα διασύνδεσης και τη διαδικασία εισόδου-εξόδου 229

ΚΕΦΑΛΑΙΟ 7

Ασκήσεις για υπερβαθμωτούς επεξεργαστές και επεξεργαστές πολύ μεγάλου μήκους εντολών 253

ΠΡΟΛΟΓΟΣ

Το βιβλίο αυτό περιέχει ένα σύνολο αναλυτικά λυμένων ασκήσεων που καλύπτουν πλήρως την ύλη του βιβλίου «Αρχιτεκτονική Υπολογιστών, Δημήτρης Νικολός, Β. Γκιούρδας Εκδοτική, 2008». Στο βιβλίο αυτό περιλαμβάνονται λυμένες και όλες οι ασκήσεις που υπάρχουν στο τέλος κάθε κεφαλαίου του ανωτέρω αναφερθέντος βιβλίου. Ο αριθμός της άσκησης στο βιβλίο δηλώνεται μέσα σε παρένθεση. Κάθε αναφορά στο βιβλίο εννοεί το βιβλίο «Αρχιτεκτονική Υπολογιστών, Δημήτρης Νικολός, Β. Γκιούρδας Εκδοτική, 2008».

Για την καλύτερη κατανόηση απαιτείται κατ' αρχήν η εμπέδωση της αντίστοιχης ύλης του βιβλίου. Πριν τη μελέτη της λύσης κάποιας άσκησης πρέπει να γίνεται προσπάθεια επίλυσής της.

Θα ήθελα να ευχαριστήσω τον Διδάκτορα κ. Γιώργο Κεραμίδα και τον φοιτητή μου κ. Νίκο Αγγελίδη που διάβασαν τις λυμένες ασκήσεις και προέβησαν σε χρήσιμες παρατηρήσεις.

Πάτρα 1 Οκτωβρίου 2009
Δημήτρης Νικολός

Ιστοσελίδα υποστήριξης

Για τους διδάσκοντες που χρησιμοποιούν το βιβλίο στην ιστοσελίδα
http://www.ceid.upatras.gr/tca-lab/nikolos_en/Arch-book.htm είναι διαθέσιμες όλες
οι διαφάνειες των λύσεων των ασκήσεων του βιβλίου.

ΚΕΦΑΛΑΙΟ

1

Ασκήσεις στη Δομή, Οργάνωση, Λειτουργία και Αξιολόγηση Υπολογιστών

Άσκηση 1.1 (1.1 βιβλίου)

Θεωρήστε έναν επεξεργαστή του οποίου οι εντολές σε επίπεδο γλώσσας μηχανής μπορούν να χωριστούν, ανάλογα με το πλήθος των κύκλων ρολογιού (clock cycles) που απαιτούνται για την εκτέλεσή τους, σε τέσσερις ομάδες. Στον Πίνακα 1.1.1 δίνεται το πλήθος των απαιτούμενων κύκλων ρολογιού για την εκτέλεση μίας εντολής κάθε ομάδας.

Πίνακας 1.1.1

Ομάδα	Χρονικοί Περίοδοι ανά Εντολή
A	1
B	2
Γ	3
Δ	5

Να υπολογίσετε το μέσο αριθμό κύκλων ρολογιού που απαιτούνται ανά εντολή κατά την εκτέλεση ενός προγράμματος, λαμβάνοντας υπόψη ότι εκτελούνται 1500 εντολές της ομάδας A, 2000 εντολές της ομάδας B, 1000 εντολές της ομάδας Γ και 500 εντολές της ομάδας Δ.

Απάντηση

Ο μέσος αριθμός κύκλων ρολογιού για την εκτέλεση μίας εντολής του προγράμματος, ΜΚΡΕ, θα ισούται με το πηλίκο της διαίρεσης του πλήθους των κύκλων ρολογιού που απαιτούνται για την εκτέλεση όλων των εντολών του προγράμματος δια του πλήθους των εντολών που εκτελούνται. Επομένως:

$$\text{ΜΚΡΕ} = (1500 \times 1 + 2000 \times 2 + 1000 \times 3 + 500 \times 5) / (1500 + 2000 + 1000 + 500) = 11000 / 5000 = 2,2$$

Άσκηση 1.2 (1.2 βιβλίου)

Έχουμε δύο υπολογιστές Y_1 και Y_2 με την ίδια αρχιτεκτονική σε επίπεδο εντολών γλώσσας μηχανής και συχνότητες λειτουργίας της ΚΜΕ 2 GHz και 2,5 GHz αντίστοιχα. Όταν εκτελείται το πρόγραμμα Α τότε ο μέσος αριθμός κύκλων ρολογιού ανά εντολή για τους Y_1 και Y_2 είναι $MKPE_1=1,2$ και $MKPE_2=1,6$ αντίστοιχα. Ποιός από τους δύο υπολογιστές εκτελεί γρηγορότερα το πρόγραμμα αυτό;

Απάντηση

Θεωρούμε ότι η επίδραση του λειτουργικού συστήματος και της διαδικασίας εισόδου/εξόδου είναι η ίδια και για τους δύο υπολογιστές και την αγνοούμε.

Εφόσον έχουμε την ίδια αρχιτεκτονική σε επίπεδο εντολών γλώσσας μηχανής, ένα πρόγραμμα θα έχει το ίδιο πλήθος εντολών και στους δύο υπολογιστές.

Έστω ότι ο χρόνος εκτέλεσης στον Y_1 και Y_2 είναι αντίστοιχα T_1 και T_2 . Τότε:

$$\begin{aligned} T_1 &= (\text{πλήθος εντολών}) \times (\text{μέσος αριθμός κύκλων ρολογιού ανά εντολή στον } Y_1) \times \\ &\quad (\text{περίοδος ρολογιού του } Y_1) = \\ &= (\text{πλήθος εντολών}) \times (\text{μέσος αριθμός κύκλων ρολογιού ανά εντολή στον } Y_1) / \\ &\quad (\text{συχνότητα λειτουργίας του } Y_1) = \\ &= (\text{πλήθος εντολών}) \times (1,2 \text{ κύκλοι ρολογιού ανά εντολή}) / 2 \text{ GHz} = \\ &= (\text{πλήθος εντολών}) \times (1,2 \text{ κύκλοι ρολογιού ανά εντολή}) / (2 \times 10^9 \text{ κύκλοι ανά sec}) = \\ &= (\text{πλήθος εντολών}) \times (1,2 \times 0,5 \times 10^{-9} \text{ sec}) = (\text{πλήθος εντολών}) \times 0,6 \times 10^{-9} \text{ sec}. \end{aligned}$$

$$\begin{aligned} T_2 &= (\text{πλήθος εντολών}) \times (\text{μέσος αριθμός κύκλων ρολογιού ανά εντολή στον } Y_2) \times \\ &\quad (\text{περίοδος ρολογιού του } Y_2) = \\ &= (\text{πλήθος εντολών}) \times (\text{μέσος αριθμός κύκλων ρολογιού ανά εντολή στον } Y_2) / \\ &\quad (\text{συχνότητα λειτουργίας του } Y_2) = \\ &= (\text{πλήθος εντολών}) \times (1,6 \text{ κύκλοι ρολογιού ανά εντολή}) / 2,5 \text{ GHz} = \\ &= (\text{πλήθος εντολών}) \times (1,6 \text{ κύκλοι ρολογιού ανά εντολή}) / (2,5 \times 10^9 \text{ κύκλοι ανά sec}) = \\ &= (\text{πλήθος εντολών}) \times (1,6 \times 0,4 \times 10^{-9} \text{ sec}) = (\text{πλήθος εντολών}) \times 0,64 \times 10^{-9} \text{ sec}. \end{aligned}$$

Επομένως ο χρόνος εκτέλεσης στον υπολογιστή Y_1 είναι μικρότερος απ' ότι στον Y_2 . Παρατηρούμε ότι δεν μπορούμε να βγάλουμε συμπέρασμα για την απόδοση ενός υπολογιστή μόνο από τη συχνότητα του ρολογιού της ΚΜΕ.

Άσκηση 1.3 (1.3 βιβλίου)

Ένας επεξεργαστής χρησιμοποιείται σε εφαρμογές στις οποίες το 30 % του χρόνου δαπανάται για την εκτέλεση πράξεων πολλαπλασιασμού και το 10% για την εκτέλεση πράξεων διαίρεσης. Κατά τη σχεδίαση ενός νέου μοντέλου οι σχεδιαστές έχουν τη δυνατότητα λόγω κόστους να υλοποιήσουν είτε ένα νέο κύκλωμα πολλαπλασιασμού, το οποίο θα διπλασιάσει την ταχύτητα των πολλαπλασιασμών, είτε ένα νέο κύκλωμα διαίρεσης, το οποίο θα τετραπλασιάσει την ταχύτητα των διαιρέσεων. Ποια επιλογή θα οδηγήσει στη μέγιστη απόδοση;

Απάντηση

Έστω ότι T_x είναι ο χρόνος εκτέλεσης των εντολών τύπου x . Τότε ο συνολικός χρόνος εκτέλεσης του προγράμματος θα είναι:

$$T = T_{\text{πολλαπλασιασμών}} + T_{\text{διαιρέσεων}} + T_{\text{άλλων εντολών}}$$

με

$$T_{\text{πολλαπλασιασμών}} = 0,3 \times T$$

$$T_{\text{διαιρέσεων}} = 0,1 \times T$$

$$T_{\text{άλλων εντολών}} = 0,6 \times T$$

(1.3.1)

Εάν αντικατασταθεί η μονάδα του πολλαπλασιασμού, τότε ο χρόνος εκτέλεσης του προγράμματος θα είναι:

$$T_{\pi} = (T_{\text{πολλαπλασιασμών}} / 2) + T_{\text{διαιρέσεων}} + T_{\text{άλλων εντολών}} \quad (1.3.2)$$

ενώ αν αντικατασταθεί η μονάδα της διαίρεσης τότε ο χρόνος εκτέλεσης του προγράμματος θα είναι:

$$T_{\delta} = T_{\text{πολλαπλασιασμών}} + (T_{\text{διαιρέσεων}} / 4) + T_{\text{άλλων εντολών}} \quad (1.3.3)$$

Λαμβάνοντας υπόψη τις σχέσεις (1.3.1), από τις σχέσεις (1.3.2) και (1.3.3) παίρνουμε:

$$T_{\pi} = (0,3 \times T / 2) + 0,1 \times T + 0,6 \times T = 0,15 \times T + 0,1 \times T + 0,6 \times T = 0,85 \times T$$

και

$$T_{\delta} = 0,3 \times T + (0,1 \times T / 4) + 0,6 \times T = 0,3 \times T + 0,025 \times T + 0,6 \times T = 0,925 \times T$$

Από τις παραπάνω σχέσεις παρατηρούμε ότι η αντικατάσταση του πολλαπλασιαστή οδηγεί στη μεγαλύτερη μείωση του χρόνου εκτέλεσης ενός προγράμματος, επομένως στη μεγαλύτερη αύξηση της απόδοσης.

Παρατηρούμε ότι καίτοι η καθυστέρηση του διαιρέτη μειώθηκε κατά δύο φορές περισσότερο απ' ό,τι του πολλαπλασιαστή, η αντικατάσταση του πολλαπλασιαστή οδήγησε στη μεγαλύτερη αύξηση της απόδοσης. Αυτό οφείλεται στο γεγονός ότι η συμβολή του πολλαπλασιαστή στο συνολικό χρόνο εκτέλεσης (30%) είναι πολύ μεγαλύτερη απ' ό,τι του διαιρέτη (10%). Επομένως θα μπορούσαμε να βγάλουμε το γενικότερο συμπέρασμα: όταν θέλουμε να αυξήσουμε την απόδοση ενός συστήματος θα πρέπει να βελτιώνουμε την απόδοση της μονάδας που βάζει τη μεγαλύτερη καθυστέρηση, συνήθως της μονάδας που χρησιμοποιείται πιο πολύ. Αυτός είναι γνωστός και ως νόμος του Amdahl.

Άσκηση 1.4 (1.4 βιβλίου)

Θεωρούμε δύο υπολογιστές Y1 και Y2 που έχουν υλοποιηθεί με τον ίδιο επεξεργαστή E, του οποίου ο χρόνος κύκλου ρολογιού του είναι 5ns. Ο επεξεργαστής E δεν περιέχει κρυφή μνήμη και το σύνολο των εντολών του σε επίπεδο γλώσσας μηχανής αποτελείται από πέντε είδη εντολών, όπως φαίνεται στον Πίνακα 1.4.1. Η κύρια μνήμη του υπολογιστή Y1 έχει υλοποιηθεί με πιο αργή μνήμη από ό,τι η κύρια μνήμη του υπολογιστή Y2. Συγκεκριμένα θεωρούμε ότι κάθε προσπέλαση της κύριας μνήμης του υπολογιστή Y1 εισάγει καθυστέρηση 30 κύκλων ρολογιού, ενώ κάθε προσπέλαση της κύριας μνήμης του υπολογιστή Y2 εισάγει καθυστέρηση 20 κύκλων ρολογιού. Υπολογίστε το χρόνο εκτέλεσης σε κάθε ένα υπολογιστή ενός προγράμματος του οποίου το πλήθος των εκτελούμενων εντολών δίνονται στον Πίνακα 1.4.2 και συγκρίνετέ τους μεταξύ τους. Υποθέστε ότι τόσο το εκτελούμενο πρόγραμμα όσο και τα απαιτούμενα δεδομένα βρίσκονται στην κύρια μνήμη. Η κύρια μνήμη έχει οργάνωση μιας ψηφιολέξης ανά θέση μνήμης και η προσπέλαση γίνεται ανά ψηφιολέξη.

Πίνακας 1.4.1

είδος εντολής	Κύκλοι ρολογιού ανά εντολή	Ψηφιολέξεις που καταλαμβάνει η εντολή στη μνήμη	Πλήθος ψηφιολέξεων μνήμης που προσπελάζει η εντολή κατά την εκτέλεσή της
A	4	1	0
B	7	1	1
Γ	7	2	1
Δ	10	3	0
E	10	3	2

Πίνακας 1.4.2

είδος εντολής	Πλήθος εκτελούμενων εντολών
A	600
B	300
Γ	10
Δ	100
E	10

Απάντηση

Από τον Πίνακα 1.4.1 γνωρίζουμε το πλήθος των θέσεων μνήμης που καταλαμβάνει κάθε εντολή και επομένως το πλήθος των προσπελάσεων της κύριας μνήμης που απαιτούνται για την προσκόμιση κάθε εντολής. Επίσης για τις εντολές που απαιτούν μεταφορά δεδομένων μεταξύ της ΚΜΕ και της κύριας μνήμης από τον Πίνακα 1.4.1 γνωρίζουμε το πλήθος των προσπελάσεων που απαιτούνται για τη μεταφορά των δεδομένων από ή προς την κύρια μνήμη.

Το πλήθος των κύκλων ρολογιού που απαιτούνται για την εκτέλεση μίας εντολής του είδους i , KPE_i , θα έχει δύο συνιστώσες. Το πλήθος των κύκλων ρολογιού που απαιτούνται για την εκτέλεση μίας εντολής του είδους i εάν θεωρήσουμε ότι το σύστημα μνήμης δεν βάζει καμία καθυστέρηση στην ανάγνωση εντολών και δεδομένων ή την εγγραφή αποτελεσμάτων, $KPE_i(KME)$, και το πλήθος κύκλων ρολογιού που απαιτούνται για την προσκόμιση της εντολής και των δεδομένων ή την ανάγνωση της εντολής και την εγγραφή των αποτελεσμάτων, $KPE_i(\text{κύρια μνήμη})$. Έστω ότι το Π_i δηλώνει το πλήθος των εντολών του είδους i που εκτελούνται. Ο χρόνος εκτέλεσης θα δίνεται από τη σχέση:

$$XE = \left\{ \sum_{i=1}^5 [(KPE_i(KME) + KPE_i(\text{κύρια μνήμη})) \times \Pi_i] \right\} \times (\text{διάρκεια κύκλου ρολογιού})$$

ή ισοδύναμα

$$XE = \left\{ \sum_{i=1}^5 [KPE_i(KME) \times \Pi_i] + \sum_{i=1}^5 [KPE_i(\text{κύρια μνήμη}) \times \Pi_i] \right\} \times (\text{διάρκεια κύκλου ρολογιού})$$

Το πλήθος των κύκλων ρολογιού $KPE_i(KME)$ για κάθε είδος εντολής δίνεται από την στήλη “κύκλοι ρολογιού ανά εντολή” του Πίνακα 1.4.1. Επομένως ο όρος

$$\sum_{i=1}^5 [KPE_i(KME) \times \Pi_i] \text{ παίρνει την ίδια τιμή για τον υπολογιστή Y1 και Y2.}$$

$$\begin{aligned}
& \sum_{i=1}^5 [(KPE_i(KME) \times \Pi_i = \\
& = (KPE_A(KME) \times \Pi_A + KPE_B(KME) \times \Pi_B + KPE_\Gamma(KME) \times \Pi_\Gamma + KPE_\Delta(KME) \times \Pi_\Delta \\
& \quad + KPE_E(KME) \times \Pi_E) = \\
& \quad = 4 \times 600 + 7 \times 300 + 7 \times 10 + 10 \times 100 + 10 \times 10 \text{ κύκλοι ρολογιού} = \\
& \quad = 2400 + 2100 + 70 + 1000 + 100 \text{ κύκλοι ρολογιού} = 5670 \text{ κύκλοι ρολογιού}
\end{aligned}$$

Η τιμή του KPE_i (κύρια μνήμη) εξαρτάται από το πλήθος των προσπελάσεων της κύριας μνήμης που απαιτούνται για την προσκόμιση της εντολής του είδους i και το πλήθος των προσπελάσεων που απαιτούνται για ανάγνωση ή εγγραφή δεδομένων που προσπελούνται από μια εντολή του είδους i και δίνεται για κάθε είδος εντολής από το άθροισμα των στηλών “Ψηφιολέξεις που καταλαμβάνει η εντολή στη μνήμη” και “Πλήθος ψηφιολέξεων μνήμης που προσπελώνει η εντολή κατά την εκτέλεσή της” του Πίνακα 1.4.1. Επομένως:

$$\begin{aligned}
& KPE_i(\text{κύρια μνήμη}) = \\
& = (\text{πλήθος προσπελάσεων κύριας μνήμης})_i \times (\text{χρόνος καθυστέρησης σε κύκλους ρολογιού})
\end{aligned}$$

Το πλήθος προσπελάσεων της κύριας μνήμης ανά εντολή δεν εξαρτάται από τον χρόνο προσπέλασης της κύριας μνήμης, επομένως έχει την ίδια τιμή για τους υπολογιστές Y1 και Y2. Άρα:

$$\begin{aligned}
& \sum_{i=1}^5 (KPE_i(\text{κύρια μνήμη}) \times \Pi_i = \\
& = \sum_{i=1}^5 (\text{πλήθος προσπελάσεων κύριας μνήμης})_i \times (\text{χρόνος καθυστέρησης ανά προσπέ-} \\
& \text{λαση σε κύκλους ρολογιού}) \times \Pi_i = \\
& = \left\{ \sum_{i=1}^5 (\text{πλήθος προσπελάσεων κύριας μνήμης})_i \times \Pi_i \right\} \times (\text{χρόνος καθυστέρησης ανά} \\
& \text{προσπέλαση σε κύκλους ρολογιού}) = \\
& = (1 \times 600 + 2 \times 300 + 3 \times 10 + 3 \times 100 + 5 \times 10) \times (\text{χρόνος καθυστέρησης ανά προσπέ-} \\
& \text{λαση σε κύκλους ρολογιού}) = 1580 \times (\text{χρόνος καθυστέρησης ανά προσπέλαση σε} \\
& \text{κύκλους ρολογιού}).
\end{aligned}$$

Επομένως

$XE = 5670 + 1580 \times (\text{χρόνος καθυστέρησης ανά προσπέλαση σε κύκλους ρολογιού})$
κύκλοι ρολογιού.

Λαμβάνοντας υπόψη ότι ο χρόνος καθυστέρησης που εισάγει η κύρια μνήμη για κάθε προσπέλασή της στον υπολογιστή Y1 και Y2 είναι 30 και 20 κύκλοι ρολογιού αντίστοιχα, παίρνουμε:

$$XE(Y1) = (5670 + 1580 \times 30) \times (\text{διάρκεια κύκλου ρολογιού}) =$$

$$= 53.070 \times 5ns = 265.350 ns = 265,35 \mu s$$

$$XE(Y2) = (5670 + 1580 \times 20) \times (\text{διάρκεια κύκλου ρολογιού}) =$$

$$= 37.270 \times 5ns = 186.350 ns = 186,35 \mu s$$

Από τα ανωτέρω συμπεραίνουμε ότι το πρόγραμμα θα εκτελεστεί πιο γρήγορα στον υπολογιστή Y2, ο οποίος έχει γρηγορότερη κύρια μνήμη.

Άσκηση 1.5 (1.5 βιβλίου)

Θεωρούμε ένα υπολογιστή Y1 που αποτελείται από την ΚΜΕ, μία ενοποιημένη κρυφή μνήμη για εντολές και δεδομένα με λόγο επιτυχίας $\lambda=0,9$ που η προσπέλασή της βάζει καθυστέρηση ενός κύκλου ρολογιού και κύρια μνήμη που σε κάθε περίπτωση αποτυχίας να βρεθεί η πληροφορία στην κρυφή μνήμη βάζει καθυστέρηση 20 κύκλων ρολογιού για τη μεταφορά ενός μπλοκ πληροφορίας από την κύρια μνήμη στην κρυφή μνήμη. Ο χρόνος κύκλου ρολογιού είναι 5ns. Υποθέστε ότι το σύνολο των εντολών του προγράμματος σε επίπεδο γλώσσας μηχανής αποτελείται από τα πέντε είδη εντολών που περιγράφονται στον Πίνακα 1.4.1 της Άσκησης 1.4. Υπολογίστε το χρόνο εκτέλεσης ενός προγράμματος του οποίου τα χαρακτηριστικά δίνονται στον Πίνακα 1.4.2 της Άσκησης 1.4. Υποθέστε ότι τόσο το εκτελούμενο πρόγραμμα όσο και τα απαιτούμενα δεδομένα βρίσκονται στην κύρια μνήμη. Η ΚΜΕ μπορεί να προσπελάσει εντολές και δεδομένα μόνο από την κρυφή μνήμη, η προσπέλαση γίνεται ανά ψηφιολέξη και τόσο η κρυφή όσο και η κύρια μνήμη έχουν οργάνωση μιας ψηφιολέξης ανά θέση μνήμης.

Απάντηση

α. Από τον Πίνακα 1.4.1 γνωρίζουμε το πλήθος των θέσεων μνήμης που καταλαμβάνει κάθε εντολή και επομένως το πλήθος των προσπελάσεων του συστήματος μνήμης που απαιτείται για την προσκόμιση κάθε εντολής. Επίσης για τις εντολές που απαιτούν μεταφορά δεδομένων μεταξύ της ΚΜΕ και της μνήμης γνωρίζουμε από τον Πίνακα 1.4.1 το πλήθος των προσπελάσεων που απαιτούνται για τη μεταφορά των δεδομένων από ή προς το σύστημα μνήμης.

Η ΚΜΕ μπορεί να προσπελάσει εντολές και δεδομένα μόνο από την κρυφή μνήμη, επομένως στην περίπτωση που η ζητούμενη πληροφορία δε βρίσκεται στην κρυφή μνήμη ο χρόνος προσπέλασης θα είναι $20+1$ κύκλοι ρολογιού = 21 κύκλοι ρολογιού.

Έστω ότι XEE_i είναι ο Χρόνος Εκτέλεσης Εντολής τύπου i , σε κύκλους ρολογιού. Τότε

$$XEE_i = (\text{κύκλοι ρολογιού για την εκτέλεση της εντολής στην ΚΜΕ}) + \\ [(\text{πλήθος ψηφιολέξεων που καταλαμβάνει η εντολή στη μνήμη}) + (\text{πλήθος} \\ \text{ψηφιολέξεων μνήμης που προσπελαύνει η εντολή κατά την εκτέλεσή της})] \times \\ [(\text{λόγος επιτυχίας της κρυφής μνήμης}) \times (\text{κύκλοι ρολογιού για την προσπέλαση} \\ \text{μιας ψηφιολέξης από την κρυφή μνήμη}) + \\ (\text{λόγος αποτυχίας της κρυφής μνήμης}) \times (\text{κύκλοι ρολογιού για την προσπέλαση} \\ \text{μιας ψηφιολέξης από το σύστημα μνήμης})]$$

$$XEE_A = 4 + 1 \times \lambda \times 1 + 1 \times (1 - \lambda) \times 21 \text{ κύκλοι ρολογιού} = 4 + 1 \times 0,9 + (1 - 0,9) \times 21 \text{ κύκλοι ρολογιού} \\ = 7 \text{ κύκλοι ρολογιού.}$$

$$XEE_B = 7 + 2 \times \lambda \times 1 + 2 \times (1 - \lambda) \times 21 \text{ κύκλοι ρολογιού} \\ = 7 + 2 \times 0,9 + 2 \times (1 - 0,9) \times 21 \text{ κύκλοι ρολογιού} = 13 \text{ κύκλοι ρολογιού}$$

$$XEE_\Gamma = 7 + 3 \times \lambda \times 1 + 3 \times (1 - \lambda) \times 21 \text{ κύκλοι ρολογιού} \\ = 7 + 3 \times 0,9 + 3 \times (1 - 0,9) \times 21 \text{ κύκλοι ρολογιού} = 16 \text{ κύκλοι ρολογιού}$$

$$XEE_\Delta = 10 + 3 \times \lambda \times 1 + 3 \times (1 - \lambda) \times 21 \text{ κύκλοι ρολογιού} \\ = 10 + 3 \times 0,9 + 3 \times (1 - 0,9) \times 21 \text{ κύκλοι ρολογιού} = 19 \text{ κύκλοι ρολογιού}$$

$$XEE_E = 10 + 5 \times \lambda \times 1 + 5 \times (1 - \lambda) \times 21 \text{ κύκλοι ρολογιού} \\ = 10 + 5 \times 0,9 + 5 \times (1 - 0,9) \times 21 \text{ κύκλοι ρολογιού} = 25 \text{ κύκλοι ρολογιού}$$

Λαμβάνοντας υπόψη το πλήθος των εντολών κάθε είδους και τη διάρκεια του κύκλου ρολογιού, συμπεραίνουμε ότι ο συνολικός χρόνος εκτέλεσης του προγράμματος θα είναι:

$$T = (600 \times XEE_A + 300 \times XEE_B + 10 \times XEE_\Gamma + 100 \times XEE_\Delta + 10 \times XEE_E) \times 5 \text{ ns} = \\ (600 \times 7 + 300 \times 13 + 10 \times 16 + 100 \times 19 + 10 \times 25) \times 5 \text{ ns} = \\ = (4200 + 3900 + 160 + 1900 + 250) \times 5 \text{ ns} = 10.410 \times 5 \text{ ns} = 52.050 \text{ ns} = 52,05 \mu\text{s}.$$

Άσκηση 1.6 (1.6 βιβλίου)

Θεωρούμε δύο υπολογιστές Y_1 και Y_2 που έχουν υλοποιηθεί με τους επεξεργαστές E_1 και E_2 αντίστοιχα που έχουν την ίδια αρχιτεκτονική σε επίπεδο εντολών γλώσσας μηχανής αλλά έχουν υλοποιηθεί με διαφορετική τεχνολογία και έχουν χρόνο κύκλου ρολογιού 5 nsec και 2,5 nsec αντίστοιχα. Η μνήμη που χρησιμοποιήθηκε για την υλοποίηση της κύριας μνήμης τόσο του Y_1 όσο και του Y_2 έχει χρόνο προσπέλασης 50 nsec. Κανένας από τους Y_1 και Y_2 δεν διαθέτει κρυφή μνήμη και το σύνολο των εντολών τους σε επίπεδο γλώσσας μηχανής αποτελείται από τρία είδη εντολών που περιγράφονται στον Πίνακα 1.6.1 Υπολογίστε το χρόνο εκτέλεσης σε κάθε ένα υπολογιστή ενός προγράμματος του οποίου το πλήθος των εκτελούμενων εντολών δίνεται στον Πίνακα 1.6.2. Να συγκρίνετε και να σχολιάσετε τα αποτελέσματα. Υποθέστε ότι τόσο το εκτελούμενο πρόγραμμα όσο και τα απαιτούμενα δεδομένα βρίσκονται στην κύρια μνήμη. Η προσπέλαση της κύριας μνήμης γίνεται ανά ψηφιολέξη.

Πίνακας 1.6.1

Είδος εντολής	Κύκλοι ρολογιού ανά εντολή	Ψηφιολέξεις που καταλαμβάνει η εντολή στη μνήμη	Πλήθος ψηφιολέξεων μνήμης που προσπελαύνει η εντολή κατά την εκτέλεσή της
A	4	1	1
B	4	2	1
Γ	10	3	2

Πίνακας 1.6.2

Είδος εντολής	Πλήθος εκτελούμενων εντολών
A	300
B	10
Γ	100

Απάντηση

Ο χρόνος προσπέλασης της Κύριας Μνήμης είναι 50 nsec, επομένως ισούται με $50/5 = 10$ κύκλους ρολογιού του Y_1 και $50/2,5 = 20$ κύκλους ρολογιού για τον Y_2 .

Έστω ότι KPE_i (KME) είναι το πλήθος κύκλων ρολογιού που δαπανώνται για την εκτέλεση της εντολής του είδους i από την Κεντρική Μονάδα Επεξεργασίας και KPE_i (KM) είναι το πλήθος των κύκλων ρολογιού που δαπανώνται για την προσπέλαση της εντολής του είδους i και των σχετικών δεδομένων από την Κύρια Μνήμη.

Τότε ο χρόνος εκτέλεσης $XE(Y1)$ του προγράμματος στον υπολογιστή $Y1$ θα είναι:

$$\begin{aligned}
 XE(Y1) &= \left\{ \sum_{i=1}^3 [(KPE_i(KME) + KPE_i(\text{κύρια μνήμη})] \times \Pi_i \right\} \times (\text{διάρκεια} \\
 &\hspace{25em} \text{κύκλου ρολογιού}(Y1)) = \\
 &= [(4+2 \times 10) \times 300 + (4+3 \times 10) \times 10 + (10+5 \times 10) \times 100] \times 5 \text{ ns} = \\
 &= [7200 + 340 + 6000] \times 5 \text{ ns} = 67.700 \text{ ns}
 \end{aligned}$$

Ο χρόνος εκτέλεσης $XE(Y2)$ του προγράμματος στον υπολογιστή $Y2$ θα είναι:

$$\begin{aligned}
 XE(Y2) &= \left\{ \sum_{i=1}^3 [(KPE_i(KME) + KPE_i(\text{κύρια μνήμη})] \times \Pi_i \right\} \times (\text{διάρκεια} \\
 &\hspace{25em} \text{κύκλου ρολογιού}(Y2)) = \\
 &= [(4+2 \times 20) \times 300 + (4+3 \times 20) \times 10 + (10+5 \times 20) \times 100] \times 2,5 \text{ ns} = \\
 &= [13200 + 640 + 11000] \times 2,5 \text{ ns} = 62.100 \text{ ns}.
 \end{aligned}$$

Παρατηρούμε ότι αν και ο χρόνος κύκλου ρολογιού του $Y2$ είναι ο μισός του χρόνου κύκλου ρολογιού του $Y1$, ο χρόνος εκτέλεσης του προγράμματος στον $Y2$ δεν είναι ο μισός. Αυτό οφείλεται στο γεγονός ότι ο χρόνος προσπέλασης της κύριας μνήμης είναι ο ίδιος και για τους δύο υπολογιστές και ένα μεγάλο μέρος του χρόνου δαπανάται για την προσπέλαση της κύριας μνήμης.

Άσκηση 1.7

Σ' ένα υπολογιστικό σύστημα υπάρχει κρυφή μνήμη μόνο εντολών με λόγο επιτυχίας 97% και χρόνο προσπέλασης 5 nsec. Ο χρόνος προσπέλασης της κύριας μνήμης είναι 60 nsec, ενώ για τη μεταφορά ενός μπλοκ πληροφορίας από την κύρια μνήμη σε ένα πλαίσιο της κρυφής μνήμης απαιτούνται 20 κύκλοι ρολογιού. Ο κύκλος ρολογιού είναι ίσος με 5 nsec. Θεωρήστε ότι η Κεντρική Μονάδα Επεξεργασίας, ΚΜΕ, μπορεί να προσπελάσει εντολές μόνο από την κρυφή μνήμη εντολών και ότι όλα τα απαιτούμενα δεδομένα και προγράμματα υπάρχουν στην κύρια μνήμη. Σε κάθε προσπέλαση, στην ΚΜΕ μεταφέρεται μια ψηφιολέξη. Το σύνολο των εντολών του συστήματος σε επίπεδο γλώσσας μηχανής αποτελείται από τέσσερα είδη εντολών, όπως φαίνεται στον Πίνακα 1.7.1. Υπολογίστε το χρόνο εκτέλεσης ενός προγράμματος του οποίου το πλήθος των εκτελούμενων εντολών δίνονται στον Πίνακα 1.7.2. Υποθέστε ότι τόσο το εκτελούμενο πρόγραμμα όσο και τα απαιτούμενα δεδομένα βρίσκονται στην κύρια μνήμη.

Πίνακας 1.7.1

Είδος εντολής	Κύκλοι ρολογιού ανά εντολή	Ψηφιολέξεις που καταλαμβάνει η εντολή στη μνήμη	Πλήθος ψηφιολέξεων μνήμης που προσπελαύνει η εντολή κατά την εκτέλεσή της
A	4	1	0
B	7	1	1
Γ	10	3	0
Δ	10	3	2

Πίνακας 1.7.2

είδος εντολής	Πλήθος εκτελούμενων εντολών
A	600
B	300
Γ	100
Δ	40

Απάντηση

α. Από τον Πίνακα 1.7.1 γνωρίζουμε το πλήθος των θέσεων μνήμης που καταλαμβάνει κάθε εντολή και επομένως το πλήθος των προσπελάσεων του συστήματος μνήμης που απαιτείται για την προσκόμιση κάθε εντολής. Επίσης για τις εντολές που απαιτούν μεταφορά δεδομένων μεταξύ της ΚΜΕ και της μνήμης από τον ίδιο Πίνακα γνωρίζουμε το πλήθος των προσπελάσεων που απαιτούνται για τη μεταφορά των δεδομένων από ή προς το σύστημα μνήμης.

Η ΚΜΕ μπορεί να προσπελάσει εντολές μόνο από την κρυφή μνήμη, επομένως στην περίπτωση που η ζητούμενη πληροφορία δε βρίσκεται στην κρυφή μνήμη θα πρέπει πρώτα να μεταφερθεί ένα μπλοκ πληροφορίας από την κύρια μνήμη στην κρυφή μνήμη και στη συνέχεια η ζητούμενη εντολή στην ΚΜΕ, άρα ο χρόνος προσπέλασης σε κύκλους θα είναι $20+1$ κύκλοι ρολογιού = 21 κύκλοι ρολογιού.

Επειδή το σύστημα δεν διαθέτει κρυφή μνήμη δεδομένων, ο χρόνος προσπέλασης μιας ψηφιολέξης δεδομένων ισούται με 60 nsec ή ισοδύναμα με 12 κύκλους ρολογιού.

Έστω ότι XEE_i είναι ο Χρόνος Εκτέλεσης Εντολής τύπου i , σε κύκλους ρολογιού.

Τότε

$$\begin{aligned}
 XEE_i = & (\text{κύκλοι ρολογιού για την εκτέλεση της εντολής στην ΚΜΕ}) + \\
 & (\text{πλήθος ψηφιολέξεων που καταλαμβάνει η εντολή στη μνήμη}) \times \\
 & [(\text{λόγος επιτυχίας της κρυφής μνήμης εντολών}) \times (\text{κύκλοι ρολογιού για την} \\
 & \text{προσπέλαση μιας ψηφιολέξης από την κρυφή μνήμη}) + \\
 & (\text{λόγος αποτυχίας της κρυφής μνήμης εντολών}) \times (\text{κύκλοι ρολογιού για την} \\
 & \text{προσπέλαση μιας ψηφιολέξης από το σύστημα μνήμης})] + \\
 & (\text{πλήθος ψηφιολέξεων μνήμης που προσπελαύνει η εντολή κατά την εκτέλεσή} \\
 & \text{της}) \times (\text{κύκλοι ρολογιού για την προσπέλαση μιας ψηφιολέξης από την κύρια} \\
 & \text{μνήμη})
 \end{aligned}$$

$$\begin{aligned}
 XEE_A = & 4 + 1 \times [\lambda \times 1 + (1 - \lambda) \times 21] + 0 \times 12 \text{ κύκλοι ρολογιού} = 4 + 1 \times [0,97 \times 1 + (1 - 0,97) \times 21] \\
 & \text{κύκλοι ρολογιού} = 5,6 \text{ κύκλοι ρολογιού.}
 \end{aligned}$$

$$\begin{aligned}
 XEE_B = & 7 + 1 \times [\lambda \times 1 + (1 - \lambda) \times 21] + 1 \times 12 \text{ κύκλοι ρολογιού} = 7 + 1 \times [0,97 \times 1 + (1 - 0,97) \times 21] + \\
 & 1 \times 12 \text{ κύκλοι ρολογιού} = 20,6 \text{ κύκλοι ρολογιού.}
 \end{aligned}$$

$$\begin{aligned}
 XEE_\Gamma = & 10 + 3 \times [\lambda \times 1 + (1 - \lambda) \times 21] + 0 \times 12 \text{ κύκλοι ρολογιού} = 10 + 3 \times [0,97 \times 1 + (1 - 0,97) \times 21] \\
 & \text{κύκλοι ρολογιού} = 14,8 \text{ κύκλοι ρολογιού.}
 \end{aligned}$$

$$\begin{aligned}
 XEE_\Delta = & 10 + 3 \times [\lambda \times 1 + (1 - \lambda) \times 21] + 2 \times 12 \text{ κύκλοι ρολογιού} = 10 + 3 \times [0,97 \times 1 + (1 - 0,97) \times 21] \\
 & + 2 \times 12 \text{ κύκλοι ρολογιού} = 38,8 \text{ κύκλοι ρολογιού.}
 \end{aligned}$$

Λαμβάνοντας υπόψη το πλήθος των εντολών κάθε είδους και τη διάρκεια του κύκλου ρολογιού, συμπεραίνουμε ότι ο συνολικός χρόνος εκτέλεσης του προγράμματος θα είναι:

$$\begin{aligned}
 T = & (600 \times XEE_A + 300 \times XEE_B + 100 \times XEE_\Gamma + 40 \times XEE_\Delta) \times 5 \text{ ns} = \\
 = & (600 \times 5,6 + 300 \times 20,6 + 100 \times 14,8 + 40 \times 38,8) \times 5 \text{ ns} = 12.572 \times 5 \text{ ns} = 62.860 \text{ ns} = \\
 = & 62,86 \mu\text{s.}
 \end{aligned}$$

ΚΕΦΑΛΑΙΟ

2

Ασκήσεις στην οργάνωση και διαχείριση της πληροφορίας στον υπολογιστή

Ασκηση 2.1 (2.1 βιβλίου)

Επιβεβαιώστε με ένα παράδειγμα ότι η αποθήκευση δεκαδικών αριθμών σε απλή BCD μορφή απαιτεί περισσότερα δυαδικά ψηφία από ότι αν μετατρέψουμε τον δεκαδικό αριθμό σε δυαδικό και αποθηκεύσουμε τον δυαδικό. Στη συνέχεια υπολογίστε πόσο επί τις εκατό περισσότερα δυαδικά ψηφία απαιτεί η αποθήκευση δεκαδικών αριθμών σε αναπαράσταση BCD από ότι σε καθαρά δυαδική αναπαράσταση.

Απάντηση

α. Θεωρώ τον δεκαδικό αριθμό 999. Για την αναπαράσταση του αριθμού αυτού στο BCD απαιτούνται $3 \times 4 = 12$ δυαδικά ψηφία, ενώ στο δυαδικό απαιτούνται $n = \lceil \log_2 999 \rceil = 10$ δυαδικά ψηφία. Παρατηρούμε ότι απαιτούνται 20% περισσότερα δυαδικά ψηφία στην περίπτωση χρησιμοποίησης BCD αναπαράστασης.

β. Ένας δεκαδικός αριθμός των k δεκαδικών ψηφίων απαιτεί $4 \times k$ δυαδικά ψηφία για την αναπαράστασή του σε BCD μορφή. Επομένως με $4 \times k$ δυαδικά ψηφία στο BCD μπορούν να παρασταθούν 10^k διαφορετικοί αριθμοί. Οι 10^k αριθμοί μπορούν να παρασταθούν με $\lceil \log_2 10^k \rceil$ δυαδικά ψηφία. Συνεπώς για να παραστήσουμε 10^k αριθμούς απαιτούνται $4 \times k$ δυαδικά ψηφία στην BCD αναπαράσταση ενώ απαιτούνται $\lceil \log_2 10^k \rceil = \lceil (\log_{10} 10^k) / (\log_{10} 2) \rceil = \lceil k/0,3 \rceil$ δυαδικά ψηφία στην δυαδική αναπαράσταση, προφανώς $4 \times k > \lceil k/0,3 \rceil$. Δηλαδή στο δυαδικό απαιτούνται

$\lceil k/0,3 \rceil$ δυαδικά ψηφία ενώ στο BCD απαιτούνται $4 \times k - \lceil k/0,3 \rceil$ περισσότερα

στα 100

απαιτούνται Z περισσότερα

$$\begin{aligned} \text{άρα } Z &= ((4 \times k - \lceil k/0,3 \rceil) / \lceil k/0,3 \rceil) \times 100 = ((4k / \lceil k/0,3 \rceil - 1) \times 100 \approx ((4k / (k/0,3) - 1) \times 100 \\ &= (4 \times 0,3 - 1) \times 100 = (1,2 - 1) \times 100 = 20\%. \end{aligned}$$

Άσκηση 2.2 (2.2 βιβλίου)

α. Να διερευνήσετε εάν είναι δυνατόν να πολλαπλασιάσουμε ένα αριθμό BCD (με βάρη 8421) με το 2 ολισθαίνοντάς τον προς τα αριστερά και τι διορθωτικές πράξεις απαιτούνται για να το πετύχουμε.

β. Να περιγράψετε αλγόριθμο μετατροπής δυαδικού αριθμού, με οποιοδήποτε αριθμό ψηφίων, σε δεκαδικό με αναπαράσταση BCD.

Απάντηση

α. Κατ' αρχήν θα πρέπει να τονίσουμε ότι ένας μονοψήφιος BCD αριθμός με βάρη 8421 έχει ακριβώς την ίδια αναπαράσταση στο δυαδικό σύστημα, επομένως δεν χρειάζεται ιδιαίτερη αντιμετώπιση. Ας θεωρήσουμε ένα τριψήφιο BCD αριθμό:

βάρη ψηφίων	$(8\ 4\ 2\ 1) \times 10^2$	$(8\ 4\ 2\ 1) \times 10^1$	$(8\ 4\ 2\ 1) \times 10^0$	
αρχική παράσταση:	0 0 1 0	0 1 0 0	0 1 0 0	→ 244 στο BCD
μετά την ολίσθηση:	0 1 0 0	1 0 0 0	1 0 0 0	→ 488 στο BCD

Παρατηρούμε ότι σ' αυτή την περίπτωση η ολίσθηση κατά μια θέση προς τα αριστερά είναι ισοδύναμη με πολλαπλασιασμό με το 2.

Ας εξετάσουμε στη συνέχεια μία άλλη περίπτωση:

βάρη ψηφίων	$(8\ 4\ 2\ 1) \times 10^2$	$(8\ 4\ 2\ 1) \times 10^1$	$(8\ 4\ 2\ 1) \times 10^0$	
αρχική παράσταση:	0 0 1 0	0 1 1 1	0 1 0 0	→ 274 στο BCD
μετά την ολίσθηση:	0 1 0 0	1 1 1 0	1 0 0 0	

Παρατηρούμε ότι τα τέσσερα δυαδικά ψηφία της αναπαράστασης του δεύτερου BCD ψηφίου που προέκυψε από την ολίσθηση αντιστοιχούν σε αριθμό μεγαλύτερο του 9, κάτι που δεν επιτρέπεται στο δεκαδικό. Η διόρθωση που πρέπει να κάνουμε είναι να προσθέσουμε 1 στο επόμενο πιο σημαντικό ψηφίο και να αφαιρέσουμε τον αριθμό 10 από την αναπαράσταση του δεύτερου BCD ψηφίου. (Σημειώνουμε ότι το πιο σημαντικό από δύο διαδοχικά BCD ψηφία έχει πάντα δέκα φορές μεγαλύτερο βάρος από το λιγότερο σημαντικό.) Πράγματι τότε:

μετά την ολίσθηση:	0 1 0 0	1 1 1 0	1 0 0 0	
διόρθωση	+ 1	-1 0 1 0		
	0 1 0 1	0 1 0 0	1 0 0 0	→ 548 στο BCD

Η διόρθωση που αναφέραμε ανωτέρω είναι ισοδύναμη με το να προσθέσουμε στην αναπαράσταση του ψηφίου που ξεπέρασε την τιμή 9 τον αριθμό 6. Προσθέτοντας τον αριθμό 6 δημιουργείται κρατούμενο το οποίο προστίθεται στο ψηφίο με την αμέσως μεγαλύτερη σημαντικότητα ενώ η τιμή του ψηφίου στο οποίο προσθέσαμε τον αριθμό 6 μειώνεται κατά 10, π.χ.

μετά την ολίσθηση:	0 1 0 0	1 1 1 0	1 0 0 0	
διόρθωση		+ 1 1 0		
	0 1 0 1	0 1 0 0	1 0 0 0	→ 548 στο BCD

Ας εξετάσουμε στη συνέχεια μία άλλη περίπτωση:

βάρη ψηφίων	$(8\ 4\ 2\ 1) \times 10^2$	$(8\ 4\ 2\ 1) \times 10^1$	$(8\ 4\ 2\ 1) \times 10^0$	
αρχική παράσταση:	0 0 1 0	0 0 1 0	1 0 0 1	→ 229 στο BCD
μετά την ολίσθηση:	0 1 0 0	0 1 0 1	0 0 1 0	→ 452 $\neq 229 \times 2 = 458$

Κατά την ολίσθηση προς τα αριστερά το πιο σημαντικό δυαδικό ψηφίο του λιγότερου σημαντικού BCD ψηφίου, που έχει την τιμή 1, πέρασε στη λιγότερο σημαντική θέση του επόμενου σε σημαντικότητα BCD ψηφίου, επομένως το βάρος του από 8×1 έγινε 1×10 . Λαμβάνοντας υπόψη ότι θέλουμε με την ολίσθηση να επιτύχουμε πολλαπλασιασμό με το 2 και ότι αντί το βάρος του να γίνει 16 έγινε 10, συμπεραίνουμε ότι πρέπει να προσθέσουμε τον αριθμό 6 στο BCD ψηφίο από το οποίο έφυγε το δυαδικό ψηφίο με την τιμή 1. Πράγματι

μετά την ολίσθηση:	0 1 0 0	0 1 0 1	0 0 1 0	
διόρθωση			+ 1 1 0	
	0 1 0 0	0 1 0 1	1 0 0 0	→ 458 στο BCD

Συνολικά μπορούμε να πούμε: εάν η δυαδική παράσταση που προέκυψε μετά από μία ολίσθηση για κάποιο BCD ψηφίο έχει τιμή μεγαλύτερη του 9 τότε προσθέτουμε σ' αυτό το BCD ψηφίο τον αριθμό 6. Επίσης εάν το λιγότερο σημαντικό δυαδικό ψηφίο της δυαδικής παράστασης που προέκυψε μετά από μία ολίσθηση για κάποιο BCD ψηφίο έχει την τιμή 1 τότε προσθέτουμε στο αμέσως μικρότερης σημαντικότητας BCD ψηφίο τον αριθμό 6.

Πρέπει να τονίσουμε ότι το κρατούμενο εξόδου που μπορεί να προκύψει από την πρόσθεση του 6 σε κάποιο BCD ψηφίο, δεν μπορεί να διαδοθεί πέραν του λιγότερου σημαντικού δυαδικού ψηφίου του αμέσως μεγαλύτερης σημαντικότητας BCD ψηφίου.

β. Για να μετατρέψουμε ένα δυαδικό αριθμό, με άλλα λόγια μία δυαδική παράσταση, των n δυαδικών ψηφίων σε BCD παράσταση μπορούμε να χρησιμοποιήσουμε ένα ολισθητή των $4 \times k$ θέσεων, $4 \times k \geq n$, στον οποίο αρχικά θέτουμε την τιμή 0.

Κάθε ομάδα των 4 δυαδικών ψηφίων του ολισθητή, ξεκινώντας από την λιγότερο σημαντική θέση, θεωρούμε ότι αποτελεί την παράσταση ενός BCD ψηφίου. Ολισθαίνουμε την δυαδική παράσταση βήμα προς βήμα προς τα αριστερά ώστε το περισσότερο σημαντικό της ψηφίο να εισέρχεται στη λιγότερο σημαντική θέση του ολισθητή. Μετά από κάθε ολίσθηση εφαρμόζουμε τον κανόνα: Εάν η δυαδική παράσταση που προέκυψε μετά από μία ολίσθηση για κάποιο BCD ψηφίο έχει τιμή μεγαλύτερη του 9 τότε προσθέτουμε σ' αυτό το BCD ψηφίο τον αριθμό 6. Επίσης εάν το λιγότερο σημαντικό δυαδικό ψηφίο της δυαδικής παράστασης που προέκυψε μετά από μία ολίσθηση για κάποιο BCD ψηφίο έχει την τιμή 1 τότε προσθέτουμε στο αμέσως μικρότερης σημαντικότητας ψηφίο τον αριθμό 6.

Η διαδικασία της μετατροπής ολοκληρώνεται μετά από n ολισθήσεις, όπου n είναι το πλήθος των δυαδικών ψηφίων του δυαδικού αριθμού.

Παράδειγμα μετατροπής του αριθμού 314 από δυαδική παράσταση σε BCD.

βάρη	$(8421) \times 10^2$	$(8421) \times 10^1$	$(8421) \times 10^0$	δυαδικός αριθμός
BCD αριθμός	0000	0000	0000	100111010
ολίσθηση	0000	0000	0001	00111010
ολίσθηση	0000	0000	0010	0111010
ολίσθηση	0000	0000	0100	111010
ολίσθηση	0000	0000	1001	11010
ολίσθηση	0000	0001	0011	1010
διόρθωση			+110	
	0000	0001	1001	
ολίσθηση	0000	0011	0011	010
διόρθωση			+110	
	0000	0011	1001	
ολίσθηση	0000	0111	0010	10
διόρθωση			+110	
	0000	0111	1000	
ολίσθηση	0000	1111	0001	0
διόρθωση		+110	+110	
	0001	0101	0111	
ολίσθηση	0010	1010	1110	
διόρθωση		+110	+110	
	0011	0001	0100	

Άσκηση 2.3 (2.3 βιβλίου)

α. Να διερευνήσετε εάν είναι δυνατόν να διαιρέσουμε ένα αριθμό BCD (με βάρη 8421) με το 2 ολισθαίνοντάς τον προς τα δεξιά και τι διορθωτικές πράξεις απαιτούνται για να το πετύχουμε.

β. Να περιγράψετε αλγόριθμο μετατροπής BCD αριθμού με οποιοδήποτε πλήθος ψηφίων σε δυαδικό αριθμό.

Απάντηση

α. Κατ' αρχήν θα πρέπει να τονίσουμε ότι ένας μονοψήφιος BCD αριθμός με βάρη 8421 έχει ακριβώς την ίδια παράσταση στο δυαδικό σύστημα. Ας θεωρήσουμε ένα διψήφιο BCD αριθμό:

βάρη ψηφίων	$(8\ 4\ 2\ 1) \times 10^1$	$(8\ 4\ 2\ 1) \times 10^0$	
αρχική παράσταση:	0 1 1 0	1 0 0 0	$\rightarrow 68$ στο BCD
μετά την ολίσθηση:	0 0 1 1	0 1 0 0	$\rightarrow 34$ στο BCD

Παρατηρούμε ότι σ' αυτή την περίπτωση η ολίσθηση προς τα δεξιά είναι ισοδύναμη με διαίρεση δια του 2.

Ας εξετάσουμε στη συνέχεια μία άλλη περίπτωση

βάρη ψηφίων	$(8\ 4\ 2\ 1) \times 10^1$	$(8\ 4\ 2\ 1) \times 10^0$	
αρχική παράσταση:	0 1 0 1	1 0 0 0	$\rightarrow 58$ στο BCD
μετά την ολίσθηση:	0 0 1 0	1 1 0 0	

Παρατηρούμε ότι τα τέσσερα λιγότερο σημαντικά δυαδικά ψηφία της παράστασης που προέκυψε από την ολίσθηση αντιστοιχούν σε αριθμό μεγαλύτερο του 9, κάτι που δεν επιτρέπεται στο δεκαδικό. Για να δούμε πως προέκυψε αυτό. Ο αρχικός μας αριθμός ήταν το $59 = 5 \times 10^1 + 9 \times 10^0 = 5 \times 10 + 9$. Κατά την ολίσθηση του λιγότερου σημαντικού δυαδικού ψηφίου του πιο σημαντικού ψηφίου, 5, προς τα δεξιά, από το βάρος 10 που είχε καταλήγει να έχει βάρος 8, ενώ θα έπρεπε, αφού θέλουμε διαίρεση με το 2 να πάρουμε την τιμή 5. Αυτό συνεπάγεται ότι κάθε φορά που κατά την ολίσθηση προς τα δεξιά ένα δυαδικό ψηφίο με τιμή 1 περνάει από ένα περισσότερο σημαντικό δεκαδικό ψηφίο στο αμέσως λιγότερο σημαντικό ψηφίο θα πρέπει να αφαιρούμε την τιμή 3. Ισοδύναμα μπορούμε να πούμε ότι κάθε φορά που μετά την ολίσθηση η τιμή του πιο σημαντικού δυαδικού ψηφίου σ' ένα BCD ψηφίο έχει την τιμή 1 τότε από αυτό το BCD ψηφίο αφαιρούμε τον αριθμό 3. Επομένως στο παραπάνω παράδειγμα έχουμε:

βάρη ψηφίων	$(8\ 4\ 2\ 1) \times 10^1$	$(8\ 4\ 2\ 1) \times 10^0$	
αρχική παράσταση:	0 1 0 1	1 0 0 0	$\rightarrow 58$ στο BCD
μετά την ολίσθηση:	0 0 1 0	1 1 0 0	
διόρθωση		- 1 1	
	0 0 1 0	1 0 0 1	$\rightarrow 29$ στο BCD

β. Γνωρίζουμε τον τρόπο με τον οποίο με διαδοχικές διαιρέσεις με το 2 μετατρέπουμε ένα ακέραιο δεκαδικό αριθμό σε δυαδικό. Επομένως, λαμβάνοντας υπόψη τον προηγούμενο τρόπο υλοποίησης της διαίρεσης με το 2 ενός BCD αριθμού, συμπεραίνουμε ότι για να μετατρέψουμε ένα BCD αριθμό, με άλλα λόγια μία BCD παράσταση, σε δυαδική παράσταση πρέπει να κάνουμε n ολισθήσεις προς τα δεξιά, όπου n είναι το πλήθος των δυαδικών ψηφίων που απαιτούνται για τη δυαδική παράσταση. Μετά από κάθε ολίσθηση το δυαδικό ψηφίο που εξέρχεται του ολισθητή είναι ένα ψηφίο του δυαδικού αριθμού. Πρώτα προκύπτει το λιγότερο σημαντικό ψηφίο. Στην παράσταση που προκύπτει από κάθε ολίσθηση κάνουμε, αν απαιτείται, τη διόρθωση που περιγράψαμε στο ερώτημα α.

Παράδειγμα μετατροπής από BCD παράσταση σε δυαδική παράσταση.

βάρη	$(8421) \times 10^2$	$(8421) \times 10^1$	$(8421) \times 10^0$		δεκαδικός αριθμός
BCD αριθμός	0101	0001	1001		519
ολίσθηση	0010	1000	1100	1 (ΛΣΨ)	
διόρθωση		-11	-11		
	0010	0101	1001		259
ολίσθηση	0001	0010	1100	1	
διόρθωση			-11		
	0001	0010	1001		129
ολίσθηση	0000	1001	0100	1	
διόρθωση		-11			
	0000	0110	0100		64
ολίσθηση	0000	0011	0010	0	32
ολίσθηση	0000	0001	1001	0	
διόρθωση			-11		
	0000	0001	0110		16
ολίσθηση	0000	0000	1011	0	
			-11		
διόρθωση	0000	0000	1000		8
ολίσθηση	0000	0000	0100	0	4
ολίσθηση	0000	0000	0010	0	2
ολίσθηση	0000	0000	0001	0	1
ολίσθηση	0000	0000	0000	1 (ΠΣΨ)	0

Άρα η δυαδική παράσταση του BCD αριθμού 519 είναι $1000000111_{(2)}$.

Άσκηση 2.4 (2.4 βιβλίου)

Να δείξετε ότι κατά την πρόσθεση δύο αριθμών που είναι σε αναπαράσταση συμπληρώματος ως προς βάση αγνοούμε το κρατούμενο εξόδου, ενώ αν είναι σε αναπαράσταση συμπληρώματος ως προς μειωμένη βάση προσθέτουμε το κρατούμενο εξόδου στο λιγότερο σημαντικό ψηφίο του αποτελέσματος.

Απάντηση

Για την απόδειξη θα θεωρήσουμε ακέραιους αριθμούς και το δυαδικό αριθμητικό σύστημα. Η γενίκευση είναι προφανής.

Αναπαράσταση συμπληρώματος ως προς 2.

Η αναπαράσταση ενός θετικού αριθμού a , με $a < 2^{v-1}$, ή αρνητικού αριθμού a , με απόλυτη τιμή $|a| \leq 2^{v-1}$, σε μορφή συμπληρώματος ως προς 2, δίνεται από την αναπαράσταση στο δυαδικό του αριθμού A όπου

$$A = a \quad \text{εάν } a \geq 0 \quad (2.4.1)$$

$$\text{και} \quad A = 2^v - |a| = 2^v + a \quad \text{εάν } a < 0 \quad (2.4.2)$$

$$\text{Από τις σχέσεις (2.4.1) και (2.4.2) προκύπτει ότι } A = a \bmod 2^v \quad (2.4.3)$$

Υποθέστε ότι θέλουμε το άθροισμα z δύο αριθμών x και y των v δυαδικών ψηφίων που είναι σε αναπαράσταση συμπληρώματος ως προς 2. Θεωρούμε ότι δεν υπάρχει υπερχείλιση, αν υπάρχει θα ανιχνευτεί (βλέπε υποενότητα 3.1.1.1 στο βιβλίο «Αρχιτεκτονική Υπολογιστών, Δημήτρης Νικολός, Β. Γκιούρδας Εκδοτική»). Για την πρόσθεση χρησιμοποιούμε ένα δυαδικό αθροιστή. Οι παραστάσεις των x και y σε μορφή συμπληρώματος ως προς 2 δίνονται αντίστοιχα από τις δυαδικές παραστάσεις στο δυαδικό των αριθμών X και Y που προκύπτουν από τις σχέσεις (2.4.1) και (2.4.2). Για να δίνεται το άθροισμα των x και y από τη δυαδική παράσταση ενός αριθμού Z θα πρέπει $Z = z \bmod 2^v$, οπότε $Z = z \bmod 2^v = (x + y) \bmod 2^v = (x \bmod 2^v + y \bmod 2^v) \bmod 2^v = (X + Y) \bmod 2^v$

Επομένως το άθροισμα των x και y ισούται με $(X + Y) \bmod 2^v$, όμως εμείς από την πρόσθεση των X και Y πήραμε το $X + Y$.

Επομένως:

$$\text{εάν } X + Y < 2^v \quad \text{τότε } Z = X + Y \quad (2.4.4)$$

$$\text{ενώ εάν } X + Y \geq 2^v \quad \text{τότε } Z = X + Y - 2^v \quad (2.4.5)$$

Επειδή $|x| \leq 2^{v-1}$ και $|y| \leq 2^{v-1}$ από τις σχέσεις (2.4.1) και (2.4.2) έχουμε $X < 2^v$ και $Y < 2^v$, επομένως $X+Y < 2^{v+1}$, δηλαδή από την πρόσθεση δεν είναι δυνατόν να προκύψει μια παράσταση με περισσότερα από $v+1$ δυαδικά ψηφία.

Επειδή το βάρος του κρατούμενου εξόδου είναι ίσο με 2^v , από τις σχέσεις (2.4.4) και (2.4.5) προκύπτει ότι αγνοώντας το κρατούμενο εξόδου έχουμε πάντα $Z = X+Y$.

Αναπαράσταση συμπληρώματος ως προς 1.

Η αναπαράσταση ενός θετικού, ή αρνητικού αριθμού a , με απόλυτη τιμή $|a| < 2^{v-1}$, σε μορφή συμπληρώματος ως προς 1, δίνεται από την αναπαράσταση στο δυαδικό του αριθμού A όπου

$$A = a \text{ εάν } a \geq 0 \quad (2.4.6)$$

$$\text{και } A = (2^v - 1) - |a| = (2^v - 1) + a \text{ εάν } a < 0 \quad (2.4.7)$$

Από τις σχέσεις (2.4.6) και (2.4.7) λαμβάνουμε $A = a \bmod (2^v - 1)$.

Υποθέστε ότι θέλουμε το άθροισμα z δύο αριθμών x και y των v δυαδικών ψηφίων που είναι σε αναπαράσταση συμπληρώματος ως προς 1. Θεωρούμε ότι δεν υπάρχει υπερχείλιση, αν υπάρχει θα ανιχνευτεί με τον τρόπο που γνωρίζουμε. Για την πρόσθεση χρησιμοποιούμε ένα δυαδικό αθροιστή. Οι παραστάσεις των x και y σε μορφή συμπληρώματος ως προς 1 δίνονται αντίστοιχα από τις δυαδικές παραστάσεις στο δυαδικό των αριθμών X και Y που προκύπτουν από τις σχέσεις (2.4.6) και (2.4.7). Για να δίνεται το άθροισμα των x και y από τη δυαδική παράσταση ενός αριθμού Z θα πρέπει $Z = z \bmod (2^v - 1)$, οπότε

$$Z = z \bmod (2^v - 1) = (x + y) \bmod (2^v - 1) = (x \bmod (2^v - 1) + y \bmod (2^v - 1)) \bmod (2^v - 1) = (X + Y) \bmod (2^v - 1)$$

Επομένως το άθροισμα των x και y ισούται με $(X + Y) \bmod (2^v - 1)$, όμως εμείς από την πρόσθεση των X και Y πήραμε το $X+Y$.

Επομένως έχουμε τις κάτωθι περιπτώσεις:

$$\text{Εάν } X+Y < 2^v \text{ τότε } Z = X+Y$$

Εάν $X+Y = 2^v - 1$ τότε $Z = X+Y - (2^v - 1)$. Όμως σε παράσταση συμπληρώματος ως προς 1 το $2^v - 1$ (το οποίο είναι v μονάδες 1...11) και το 0...00 είναι δύο αναπαραστάσεις του μηδενός, επομένως δεν χρειάζεται να κάνουμε την αφαίρεση.

$$\text{Εάν } X+Y > 2^v - 1 \text{ τότε } Z = X+Y - (2^v - 1) = X+Y - 2^v + 1$$

Επειδή $|x| < 2^{v-1}$ και $|y| < 2^{v-1}$ από τις σχέσεις (2.4.6) και (2.4.7) έχουμε $X < 2^v$ και $Y < 2^v$, επομένως $X+Y < 2^{v+1}$, δηλαδή από την πρόσθεση δεν είναι δυνατόν να προκύψει μια παράσταση με περισσότερα από $v+1$ δυαδικά ψηφία.

Επειδή το βάρος του κρατούμενου εξόδου είναι ίσο με 2^v , από την παραπάνω ανάλυση προκύπτει ότι όταν υπάρχει κρατούμενο εξόδου εάν το αγνοήσουμε και προσθέσουμε τη μονάδα στο αποτέλεσμα έχουμε σε όλες τις περιπτώσεις $Z = X+Y$ ή με άλλα λόγια προσθέτοντας το κρατούμενο εξόδου στο λιγότερο σημαντικό δυαδικό ψηφίο του αποτελέσματος παίρνουμε $Z = X+Y$.

Άσκηση 2.5 (2.5 βιβλίου)

Να υπολογίσετε το μέγιστο σφάλμα σε αναπαράσταση σταθερής υποδιαστολής και αριθμητικό σύστημα με βάση το β , όταν έχουμε k κλασματικά ψηφία, για τις ακόλουθες δύο περιπτώσεις:

- α. χρησιμοποιείται η μέθοδος της περικοπής
- β. χρησιμοποιείται η μέθοδος της στρογγυλοποίησης.

Απάντηση

Υποθέστε ότι έχουμε ένα αριθμητικό σύστημα με βάση το β , τότε τα ψηφία a_i θα ικανοποιούν τη σχέση $0 \leq a_i \leq \beta-1$.

Έστω $N = a_{\mu}a_{\mu-1}\dots a_1a_0, a_{-1}a_{-2}a_{-3}a_{-4}a_{-5}a_{-6}\dots a_{-k+1}a_{-k}a_{-k-1}\dots$ ένας αριθμός σε αναπαράσταση σταθερής υποδιαστολής.

α. Εάν θέλουμε να κρατήσουμε μόνο k ψηφία δεξιά της υποδιαστολής εφαρμόζοντας την τεχνική της αποκοπής η παράσταση N' που θα προκύψει θα είναι:

$N' = a_{\mu}a_{\mu-1}\dots a_1a_0, a_{-1}a_{-2}a_{-3}a_{-4}a_{-5}a_{-6}\dots a_{-k+1}a_{-k}$, οπότε το σφάλμα θα δίνεται από την σχέση $N-N' = 0,00\dots 0a_{-k-1}a_{-k-2}a_{-k-3}\dots$

Η παράσταση αυτή παίρνει τη μέγιστη τιμή όταν $a_j = \beta-1$ για $j = -k-1, -k-2, \dots$

Τότε το σφάλμα θα ισούται με

$$\Sigma = (\beta-1)\beta^{-k-1} + (\beta-1)\beta^{-k-2} + (\beta-1)\beta^{-k-3} + \dots = (\beta-1)\beta^{-k-1}(1 + 1/\beta + 1/\beta^2 + \dots)$$

Λαμβάνοντας υπόψη ότι το άθροισμα των άπειρων όρων φθίνουσας γεωμετρικής προόδου δίνεται από την σχέση $a/(1-\omega)$, όπου a είναι ο πρώτος όρος και ω ο λόγος της προόδου, παίρνουμε:

$$\Sigma = (\beta-1)\beta^{-k-1}(1/(1-1/\beta)) = \beta^{-k}.$$

β. Εάν θέλουμε να κρατήσουμε μόνο κ ψηφία δεξιά της υποδιαστολής εφαρμόζοντας την τεχνική της στρογγυλοποίησης, θα πρέπει να προσθέσουμε πρώτα στον N τον αριθμό $\beta^{\kappa}/2$ και στη συνέχεια να κρατήσουμε μόνο τα κ δεξιά της υποδιαστολής πιο σημαντικά ψηφία του αριθμού που θα προκύψει.

Τότε έχουμε δύο περιπτώσεις στις οποίες μεγιστοποιείται το σφάλμα.

$$\beta.1. N = \alpha_{\mu}\alpha_{\mu-1}\dots\alpha_1\alpha_0,\alpha_{-1}\alpha_{-2}\alpha_{-3}\alpha_{-4}\alpha_{-5}\alpha_{-6}\dots\alpha_{-\kappa+1}\alpha_{-\kappa}[(\beta/2)-1](\beta-1)(\beta-1)(\beta-1)(\beta-1)\dots$$

Σ' αυτή την περίπτωση μετά την πρόσθεση του $\beta^{\kappa}/2$ θα πάρουμε

$$N' = \alpha_{\mu}\alpha_{\mu-1}\dots\alpha_1\alpha_0,\alpha_{-1}\alpha_{-2}\alpha_{-3}\alpha_{-4}\alpha_{-5}\alpha_{-6}\dots\alpha_{-\kappa+1}\alpha_{-\kappa}(\beta-1)(\beta-1)(\beta-1)(\beta-1)(\beta-1)\dots,$$

και η παράσταση N'' που θα προκύψει μετά την περικοπή θα είναι:

$$N'' = \alpha_{\mu}\alpha_{\mu-1}\dots\alpha_1\alpha_0,\alpha_{-1}\alpha_{-2}\alpha_{-3}\alpha_{-4}\alpha_{-5}\alpha_{-6}\dots\alpha_{-\kappa+1}\alpha_{-\kappa}.$$

Τότε το σφάλμα δίνεται από την σχέση

$$N-N'' = 0,\alpha_{-1}\alpha_{-2}\alpha_{-3}\alpha_{-4}\alpha_{-5}\alpha_{-6}\dots\alpha_{-\kappa+1}\alpha_{-\kappa}((\beta/2)-1)(\beta-1)(\beta-1)(\beta-1)(\beta-1)\dots,$$

με $\alpha_i = 0$ για $i = -1, -2, \dots, -\kappa$.

$$\text{Οπότε το σφάλμα θα ισούται με } \Sigma = ((\beta/2)-1)\beta^{-\kappa-1} + (\beta-1)\beta^{-\kappa-2} + (\beta-1)\beta^{-\kappa-3} + \dots = ((\beta/2)-1)\beta^{-\kappa-1} + (\beta-1)\beta^{-\kappa-2}(1 + 1/\beta + 1/\beta^2 + \dots)$$

Λαμβάνοντας υπόψη το άθροισμα των άπειρων όρων φθίνουσας γεωμετρικής προόδου παίρνουμε $\Sigma = ((\beta/2)-1)\beta^{-\kappa-1} + \beta^{-\kappa-1} = \beta^{-\kappa}/2$

$$\beta.2. N = \alpha_{\mu}\alpha_{\mu-1}\dots\alpha_1\alpha_0,\alpha_{-1}\alpha_{-2}\alpha_{-3}\alpha_{-4}\alpha_{-5}\alpha_{-6}\dots\alpha_{-\kappa+1}\alpha_{-\kappa}(\beta/2)0000000\dots$$

Σ' αυτή την περίπτωση μετά την πρόσθεση του $\beta^{\kappa}/2$ θα πάρουμε

$$N' = \alpha_{\mu}\alpha_{\mu-1}\dots\alpha_1\alpha_0,\alpha_{-1}\alpha_{-2}\alpha_{-3}\alpha_{-4}\alpha_{-5}\alpha_{-6}\dots\alpha_{-\kappa+1}\alpha'_{-\kappa}0000000\dots \text{ με } \alpha'_{-\kappa} = \alpha_{-\kappa} + 1 \text{ (χωρίς να βλάπτει την γενικότητα θεωρήσαμε ότι } \alpha_{-\kappa} < \beta-1), \text{ οπότε το σφάλμα θα δίνεται από την σχέση } N' - N = 0,\alpha_{-1}\alpha_{-2}\alpha_{-3}\alpha_{-4}\alpha_{-5}\alpha_{-6}\dots\alpha_{-\kappa}(\beta/2)0000000\dots \text{ με } \alpha_i = 0 \text{ για } i = -1, -2, \dots, -\kappa. \text{ Τότε το σφάλμα θα ισούται με}$$

$$\text{και θα είναι ίσο με } \Sigma = (\beta/2)\beta^{-\kappa-1} = \beta^{-\kappa}/2.$$

Παρατηρούμε ότι το μέγιστο σφάλμα στην περίπτωση χρησιμοποίησης της στρογγυλοποίησης είναι το μισό από το μέγιστο σφάλμα που μπορεί να προκύψει εφαρμόζοντας την τεχνική της περικοπής.

Άσκηση 2.6 (2.6 βιβλίου)

Δίνονται οι παρακάτω αριθμοί κινητής υποδιαστολής σύμφωνα με το πρότυπο IEEE 754:

α. 0 10000010 110000000000000000000000

β. 0 10000111 100100000000000000000000

γ. 1 11001100 110000000000000000000000

Να υπολογίσετε ποιος είναι κάθε αριθμός στη δεκαδική μορφή που χρησιμοποιούμε στην καθημερινή μας ζωή (π.χ. 9.85×10^{-3}).

Απάντηση

α.

Ο εκθέτης έχει την τιμή $E=130$, άρα δεν έχει ούτε τη μέγιστη τιμή, 255, αλλά ούτε και την ελάχιστη τιμή 0. Επομένως η συγκεκριμένη αναπαράσταση είναι η αναπαράσταση ενός κανονικοποιημένου αριθμού. Τότε η τιμή του αριθμού δίνεται από τη σχέση $N = (-1)^P \times 2^{E-127} \times (1. \Sigma_K)$.

Το κλασματικό μέρος του συντελεστή έχει την τιμή 11, επομένως $1.\Sigma_K = 1,11_{(2)} = 1,75_{(10)}$ και $N = (-1)^0 \times 2^{130-127} \times 1,75_{(10)} = 2^3 \times 1,75_{(10)} = 8 \times 1,75_{(10)} = 14_{(10)}$.

β.

Ο εκθέτης έχει την τιμή $E=135$, άρα δεν έχει ούτε τη μέγιστη τιμή, 255, αλλά ούτε και την ελάχιστη τιμή 0. Επομένως η συγκεκριμένη αναπαράσταση είναι η αναπαράσταση ενός κανονικοποιημένου αριθμού. Τότε η τιμή του αριθμού δίνεται από τη σχέση $N = (-1)^P \times 2^{E-127} \times (1. \Sigma_K)$.

Το κλασματικό μέρος του συντελεστή έχει την τιμή 1001, επομένως $1.\Sigma_K = 1,1001_{(2)}$ και $N = (-1)^0 \times 2^{135-127} \times 1,1001_{(2)} = 2^8 \times 1,1001_{(2)} = 2^4 \times 11001_{(2)} = 2^4 \times 25_{(10)} = 400$.

γ.

Ο εκθέτης έχει την τιμή $E=204$, άρα δεν έχει ούτε τη μέγιστη τιμή, 255, αλλά ούτε και την ελάχιστη τιμή 0. Επομένως η συγκεκριμένη αναπαράσταση είναι η αναπαράσταση ενός κανονικοποιημένου αριθμού. Τότε η τιμή του αριθμού δίνεται από τη σχέση $N = (-1)^P \times 2^{E-127} \times (1. \Sigma_K)$.

Το κλασματικό μέρος του συντελεστή έχει την τιμή 11, επομένως $1.\Sigma_K = 1,11_{(2)}$ και $N = (-1)^1 \times 2^{204-127} \times 1,11_{(2)} = -2^{77} \times 1,11_{(2)} = -2^{75} \times 111_{(2)} = -2^{75} \times 7_{(10)} \approx -2,6 \times 10^{23}$.

Άσκηση 2.7 (2.7 βιβλίου)

Θεωρήστε το πρότυπο IEEE 754 για αναπαράσταση αριθμών κινητής υποδιαστολής απλής ακρίβειας. Ποιους αριθμούς αντιπροσωπεύουν οι ακόλουθες αναπαραστάσεις;

- i. 0 00000000 000000000000000000000000
- ii. 1 00000000 000000000000000000000000
- iii. 0 11111111 000000000000000000000000
- iv. 1 11111111 000000000000000000000000
- v. 0 11111111 00111110100000000001000
- vi. 0 10000000 100000000000000000000000
- vii. 0 00000000 010000000000000000000000

Απάντηση

- i. 0 00000000 000000000000000000000000
- ii. 1 00000000 000000000000000000000000

Παρατηρούμε ότι και στις δύο αναπαραστάσεις, τόσο ο εκθέτης όσο και το κλασματικό μέρος του συντελεστή έχουν την τιμή 0, επομένως και οι δύο είναι αναπαραστάσεις του 0. Σύμφωνα με το πρότυπο IEEE 754 έχουμε δύο αναπαραστάσεις του μηδενός. Στην πρώτη το πιο σημαντικό ψηφίο της αναπαράστασης, που δηλώνει το πρόσημο του αριθμού, έχει την τιμή μηδέν, ενώ στη δεύτερη έχει την τιμή 1.

- iii. 0 11111111 000000000000000000000000
- iv. 1 11111111 000000000000000000000000

Παρατηρούμε ότι και στις δύο αναπαραστάσεις ο εκθέτης έχει τη μέγιστη τιμή 255. Επίσης και στις δύο αναπαραστάσεις το κλασματικό μέρος του συντελεστή έχει τη μηδενική τιμή. Στην πρώτη το πιο σημαντικό ψηφίο της αναπαράστασης, που δηλώνει το πρόσημο του αριθμού, έχει την τιμή μηδέν, ενώ στη δεύτερη έχει την τιμή 1. Επομένως η πρώτη είναι η αναπαράσταση του + άπειρο και η δεύτερη του – άπειρο.

- v. 0 11111111 00111110100000000001000

Παρατηρούμε ότι ο εκθέτης έχει τη μέγιστη τιμή 255 ενώ το κλασματικό μέρος του συντελεστή έχει μη μηδενική τιμή. Επομένως η αναπαράσταση δεν αντιστοιχεί σε κάποιο αριθμό (NaN).

vi. 0 10000000 100000000000000000000000

Ο εκθέτης έχει την τιμή $E=128$, άρα δεν έχει ούτε την μέγιστη τιμή, 255, αλλά ούτε και την ελάχιστη τιμή 0. Επομένως η συγκεκριμένη αναπαράσταση είναι η αναπαράσταση ενός κανονικοποιημένου αριθμού. Τότε η τιμή του αριθμού δίνεται από τη σχέση

$$N = (-1)^P \times 2^{E-127} \times (1. \Sigma_K)$$

Το κλασματικό μέρος του συντελεστή έχει την τιμή 1, επομένως $1. \Sigma_K = 1,1_{(2)} = 1,5_{(10)}$ και $N = (-1)^0 \times 2^{128-127} \times 1,5_{(10)} = 2 \times 1,5_{(10)} = 3_{(10)}$.

vii. 0 00000000 010000000000000000000000

Παρατηρούμε ότι ο εκθέτης έχει την ελάχιστη τιμή 0, ενώ το κλασματικό μέρος του συντελεστή έχει μη μηδενική τιμή. Επομένως η αναπαράσταση αντιστοιχεί σε μη κανονικοποιημένο αριθμό. Τότε η τιμή του αριθμού δίνεται από τη σχέση

$$N = (-1)^P \times 2^{-126} \times (0. \Sigma_K) \text{ οπότε}$$

$$N = (-1)^0 \times 2^{-126} \times 0,25_{(10)} = 2^{-126} \times 0,25_{(10)}.$$

Άσκηση 2.8

Θεωρήστε ένα υπολογιστή του οποίου το σύνολο των εντολών σε επίπεδο γλώσσας μηχανής αποτελείται από 200 εντολές, με σταθερό μήκος εντολών των 32 δυαδικών ψηφίων και 64 καταχωρητές με μήκος 32 δυαδικών ψηφίων ο καθένας.

α. Να σχεδιάσετε τη μορφή των εντολών σε επίπεδο γλώσσας μηχανής για κάθε μία από τις ακόλουθες εντολές σε συμβολική γλώσσα.

LOAD R, A	/ $R \leftarrow A$
STORE A, R	/ $A \leftarrow R$
ADD R1, R2, R3	/ $R1 \leftarrow R2 + R3$

β. Ποιό είναι το πλήθος των θέσεων μνήμης που μπορεί να διευθυνσιοδοτηθεί από τις παραπάνω εντολές;

Απάντηση

Η εντολή «LOAD R, A» ή «STORE A, R» σε επίπεδο γλώσσας μηχανής θα αποτελείται από τρία πεδία. Ένα πεδίο αποτελεί ο κωδικός λειτουργίας της εντολής, ένα άλλο πεδίο χρησιμοποιείται για τη δήλωση του καταχωρητή R και το άλλο πεδίο χρησιμοποιείται για τη δήλωση της διεύθυνσης μνήμης στην οποία είναι αποθηκευμένο το δεδομένο.

Η εντολή «ADD R1, R2, R3» σε επίπεδο γλώσσας μηχανής θα αποτελείται από τέσσερα πεδία. Ένα πεδίο αποτελεί ο κωδικός λειτουργίας της εντολής, και κάθε ένα από τα άλλα τρία πεδία χρησιμοποιείται για τη δήλωση ενός από τους καταχωρητές R1, R2 και R3.

Επειδή το σύνολο των εντολών σε επίπεδο γλώσσας μηχανής αποτελείται από 200 εντολές, ο κωδικός λειτουργίας πρέπει να είναι των $\lceil \log_2 200 \rceil = 8$ δυαδικών ψηφίων. Λαμβάνοντας υπόψη ότι ο υπολογιστής μας διαθέτει 64 καταχωρητές συμπεραίνουμε ότι για να δηλώσουμε ένα καταχωρητή απαιτούνται 6 δυαδικά ψηφία.

α. Τώρα γνωρίζουμε το πλήθος των πεδίων από τα οποία αποτελείται κάθε εντολή σε επίπεδο γλώσσας μηχανής, αλλά και το εύρος κάθε πεδίου. Το ερώτημα που προκύπτει είναι σε ποια θέση θα τοποθετηθεί κάθε πεδίο. Ο κωδικός λειτουργίας τοποθετείται σε όλες τις εντολές στην ίδια θέση, συνήθως στην αρχή (αριστερά) ή στο τέλος (δεξιά) κάθε εντολής. (Εμείς εδώ θεωρούμε ότι ο κωδικός λειτουργίας καταλαμβάνει ένα πεδίο. Σε κάποιους υπολογιστές ο κωδικός λειτουργίας καταλαμβάνει περισσότερα πεδία. Ως παράδειγμα, αναφέρουμε την περίπτωση στην οποία το ένα πεδίο δηλώνει το είδος της εντολής και το άλλο πεδίο τη συγκεκριμένη εντολή.) Επίσης, όταν τα πεδία που χρησιμοποιούνται για να δηλώσουν καταχωρητές καταλαμβάνουν τις ίδιες θέσεις σε όλες τις εντολές, τότε η υλοποίηση της μονάδας ελέγχου είναι πιο απλή. Στη συνέχεια θα δώσουμε μια πιθανή μορφή των εντολών που μας ζητάει η άσκηση σε επίπεδο γλώσσας μηχανής.

ΠΣΨ						ΛΣΨ	
31	-	24	23	-	18	17	0
ΚΛ			R		A		

Κωδικός
λειτουργίας,
ΚΛ, 8 δυαδικών
ψηφίων

Αριθμός
αναφοράς
καταχωρητή,
6 δυαδικά
ψηφία

Διεύθυνση θέσης μνήμης

Σχήμα 2.8.1 Μορφή εντολής «LOAD R, A» ή «STORE A, R» σε επίπεδο γλώσσας μηχανής. ΠΣΨ και ΛΣΨ: δηλώνουν το περισσότερο και το λιγότερο σημαντικό ψηφίο αντίστοιχα.

ΠΣΨ													ΛΣΨ	
31	-	24	23	-	18	17	-	12	11	-	6	5	-	0
ΚΛ			R2			R3			R1			X		

Κωδικός λειτουργίας, ΚΛ, 8 δυαδικών ψηφίων	Αριθμός αναφοράς καταχωρητή, 6 δυαδικά ψηφία	Αριθμός αναφοράς καταχωρητή, 6 δυαδικά ψηφία	Αριθμός αναφοράς καταχωρητή, 6 δυαδικά ψηφία	Αυτό το πεδίο δεν χρησιμοποιείται στη συγκεκριμένη εντολή
-----------------------------------------------------	----------------------------------------------------------	----------------------------------------------------------	----------------------------------------------------------	-----------------------------------------------------------------

Σχήμα 2.8.2 Μορφή εντολής «ADD R1, R2, R3» σε επίπεδο γλώσσας μηχανής.

Οι εντολές σε συμβολική γλώσσα μεταφράζονται από το συμβολομεταφραστή (assembler) σε εντολές γλώσσας μηχανής. Επομένως, η σειρά με την οποία εμφανίζονται τα τελούμενα (τελούμενα είναι οι αριθμοί καταχωρητών, οι διευθύνσεις και τα δεδομένα που εμφανίζονται σε μία εντολή) σε μια εντολή συμβολικής γλώσσας, μπορεί να είναι διαφορετική της σειράς με την οποία εμφανίζονται τα αντίστοιχα πεδία στην εντολή γλώσσας μηχανής. Ως παράδειγμα αναφέρουμε τις εντολές «STORE A, R» και «ADD R1, R2, R3» και τη μορφή των αντίστοιχων εντολών γλώσσας μηχανής των σχημάτων 2.8.1 και 2.8.2.

β. Από το σχήμα 2.8.1 παρατηρούμε ότι για τη δήλωση της διεύθυνσης διατίθενται 18 δυαδικά ψηφία. Επομένως το πλήθος των θέσεων μνήμης που μπορούν να διευθυνσιοδοτηθούν από τις εντολές «LOAD R, A» και «STORE A, R» είναι 2^{18} θέσεις μνήμης = 2^8 Κθέσεις μνήμης = 256 Κθέσεις μνήμης.

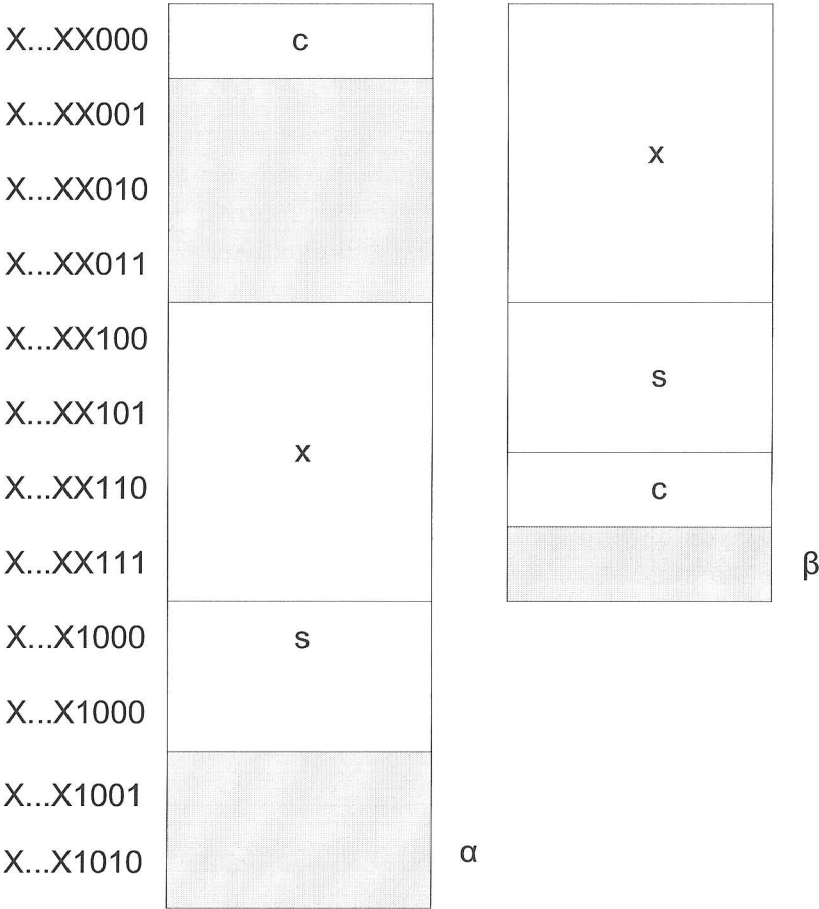
Άσκηση 2.9

Θεωρήστε ένα υπολογιστή που τα δεδομένα αποθηκεύονται στη μνήμη ευθυγραμμισμένα (aligned) και οι δομές τελειώνουν στα όρια μιας λέξης (1 λέξη = 4 ψηφιολέξεις). Ένας προγραμματιστής χρησιμοποιεί στη C την ακόλουθη δομή:

struct SI {char c; int x; short s;} example; Να αναφέρετε τα μειονεκτήματα που προκύπτουν από τον τρόπο ορισμού της δομής και να προτείνεται τρόπο μείωσης τους. (char: 8, int: 32 και short: 16 δυαδικά ψηφία)

Απάντηση

Το μειονέκτημα της δομής που χρησιμοποιεί ο προγραμματιστής είναι ότι δεσμεύονται πέντε θέσεις μνήμης χωρίς να χρησιμοποιούνται (σχήμα 2.9.1.α). Η δομή struct SI { int x; short s; char c;} examplel; δεσμεύει λιγότερο αχρησιμοποίητο χώρο (σχήμα 2.9.1.β), μόνο μία θέση.



Σχήμα 2.9.1 Ο χώρος που δεσμεύεται στη μνήμη.

Άσκηση 2.10

Θεωρούμε δύο υπολογιστές Y_1 και Y_2 οι οποίοι διαθέτουν αρτηρία δεδομένων των 32 δυαδικών ψηφίων και η μνήμη είναι οργανωμένη έτσι ώστε κάθε διεύθυνση να αντιστοιχεί σε μία ψηφιολέξη (byte), αλλά σε μια προσπέλαση να μπορούν να διαβαστούν ή να γραφούν έως και τέσσερις ψηφιολέξεις με διευθύνσεις δ , $\delta+1$, $\delta+2$ και $\delta+3$, όπου το δ είναι ακέραιο πολλαπλάσιο του τέσσερα. Τόσο στον υπολογιστή Y_1 όσο και στον Y_2 οι εντολές είναι των 32 δυαδικών ψηφίων και είναι ευθυγραμμισμένες (aligned). Στον υπολογιστή Y_1 είναι και τα δεδομένα ευθυγραμμισμένα, ενώ στον Y_2 τα δεδομένα δεν είναι κατ' ανάγκη ευθυγραμμισμένα. Συγκεκριμένα, στον Y_2 τα δεδομένα των τεσσάρων ψηφιολέξεων δεν είναι ευθυγραμμισμένα στο 70% των περιπτώσεων, ενώ τα δεδομένα των δύο ψηφιολέξεων δεν είναι ευθυγραμμισμένα στο 40% των περιπτώσεων. Και οι δύο υπολογιστές έχουν τα ίδια σύνολα εντολών γλώσσας μηχανής. Οι εντολές διακρίνονται σε τέσσερις κατηγορίες A, B, Γ, και Δ τα χαρακτηριστικά των οποίων δίνονται στις πρώτες δύο στήλες του Πίνακα 2.10.1.

Πίνακας 2.10.1

είδος εντολής	πλήθος ψηφιολέξεων δεδομένων που προσπελαύνει η εντολή	πλήθος εντολών που εκτελούνται
A	0	500
B	ένα δεδομένο της 1 ψηφιολέξης	1000
Γ	δύο δεδομένα των 2 ψηφιολέξεων το κάθε ένα	600
Δ	ένα δεδομένο των 4 ψηφιολέξεων	2000

Στους υπολογιστές Y_1 και Y_2 εκτελείται ένα πρόγραμμα τα χαρακτηριστικά του οποίου δίνονται στην τρίτη στήλη του Πίνακα 2.10.1. Εάν ο μέσος χρόνος προσπέλασης του συστήματος μνήμης είναι 4 κύκλοι ρολογιού, να υπολογίσετε το πλήθος των κύκλων ρολογιού που δαπανάται για την προσπέλαση της μνήμης κατά την εκτέλεση του προαναφερθέντος προγράμματος σε κάθε ένα από τους δύο υπολογιστές, να συγκρίνετε τα αποτελέσματα και να τα δικαιολογήσετε.

Απάντηση

Καταρχήν θα υπολογίσουμε τους κύκλους ρολογιού T_1 που δαπανώνται για την προσπέλαση της μνήμης κατά την εκτέλεση του προγράμματος στον υπολογιστή Y_1 . Στον υπολογιστή Y_1 τόσο οι εντολές όσο και τα δεδομένα είναι ευθυγραμμισμένα. Επομένως για την προσπέλαση μιας εντολής ή ενός δεδομένου απαιτείται μόνο μία προσπέλαση. Οπότε

$$\begin{aligned} T_1 &= [(\text{πλήθος εντολών είδους Α}) \times 1 + \\ &\quad (\text{πλήθος εντολών είδους Β}) \times [1 + (\text{πλήθος δεδομένων που προσπελαύνει η εντολή του} \\ &\quad \text{είδους Β})] + \\ &\quad (\text{πλήθος εντολών είδους Γ}) \times [1 + (\text{πλήθος δεδομένων που προσπελαύνει η εντολή του} \\ &\quad \text{είδους Γ})] + \\ &\quad (\text{πλήθος εντολών είδους Δ}) \times [1 + (\text{πλήθος δεδομένων που προσπελαύνει η εντολή του} \\ &\quad \text{είδους Δ})]] \times 4 \text{ κύκλοι ρολογιού} = \\ &= [500 \times [1+0] + (1000) \times (1+1) + (600) \times (1+2) + (2000) \times (1+1)] \times 4 \text{ κύκλοι ρολογιού} = \\ &= 8.300 \times 4 \text{ κύκλοι ρολογιού} = 33.200 \text{ κύκλοι ρολογιού}. \end{aligned}$$

Στον υπολογιστή Y_2 οι εντολές είναι ευθυγραμμισμένες, όμως τα δεδομένα δεν είναι κατ' ανάγκη ευθυγραμμισμένα. Πρέπει να σημειώσουμε ότι για την προσπέλαση ενός δεδομένου μιας ψηφιολέξης απαιτείται να προσπελάσουμε τη μνήμη μόνο μία φορά, είτε το δεδομένο είναι ευθυγραμμισμένο είτε όχι. Αντίθετα, όταν το δεδομένο είναι 2 ή 4 ψηφιολέξεων, τότε εάν είναι ευθυγραμμισμένο για την προσπέλασή του απαιτείται μόνο μία προσπέλαση της μνήμης, ενώ εάν δεν είναι ευθυγραμμισμένο απαιτούνται δύο προσπελάσεις της μνήμης. Οι κύκλοι ρολογιού T_2 που δαπανώνται για την προσπέλαση της μνήμης κατά την εκτέλεση του προγράμματος στον υπολογιστή Y_2 είναι:

$$\begin{aligned} T_1 &= [(\text{πλήθος εντολών είδους Α}) \times 1 + \\ &\quad (\text{πλήθος εντολών είδους Β}) \times [1 + (\text{πλήθος δεδομένων που προσπελαύνει η εντολή του} \\ &\quad \text{είδους Β})] + \\ &\quad (\text{πλήθος εντολών είδους Γ}) \times [1 + (1-0,40) \times (\text{πλήθος δεδομένων που προσπελαύνει η} \\ &\quad \text{εντολή του είδους Γ}) \times 1 + (0,40) \times (\text{πλήθος δεδομένων που προσπελαύνει η εντολή} \\ &\quad \text{του είδους Γ}) \times 2] + \\ &\quad (\text{πλήθος εντολών είδους Δ}) \times [1 + (1-0,70) \times (\text{πλήθος δεδομένων που προσπελαύνει η} \\ &\quad \text{εντολή του είδους Δ}) \times 1 + (0,70) \times (\text{πλήθος δεδομένων που προσπελαύνει η εντολή} \\ &\quad \text{του είδους Δ}) \times 2]] \times 4 \text{ κύκλοι ρολογιού} = \\ &= [500 \times 1 + (1000) \times (1+1) + 600 \times (1+0,60 \times 2 \times 1 + 0,40 \times 2 \times 2) + \\ &\quad 2000 \times (1+0,30 \times 1 \times 1 + 0,70 \times 1 \times 2)] \times 4 \text{ κύκλοι ρολογιού} = \\ &= (500+2.000+2.280+5.400) \times 4 \text{ κύκλοι ρολογιού} = 10.180 \times 4 \text{ κύκλοι ρολογιού} = \\ &= 40.720 \text{ κύκλοι ρολογιού}. \end{aligned}$$

Παρατηρούμε ότι στον υπολογιστή Y_2 , στον οποίο τα δεδομένα δεν είναι κατ' ανάγκη ευθυγραμμισμένα, απαιτούνται περισσότεροι κύκλοι ρολογιού για την προσπέλαση της μνήμης. Αυτό οφείλεται στο γεγονός ότι για την προσπέλαση ενός μη ευθυγραμμισμένου δεδομένου των 2 ή 4 ψηφιολέξεων απαιτούνται δύο προσπελάσεις της μνήμης.

Άσκηση 2.11

Ποιές είναι οι συναρτήσεις που υλοποιούν τα τμήματα προγραμμάτων σε συμβολική γλώσσα που ακολουθούν;

α) Αρχιτεκτονική που βασίζεται στη χρήση μηχανισμού στοίβας (stack architecture):

```
PUSH A
PUSH B
PUSH C
ADD
PUSH D
ADD
MUL
PUSH A
ADD
POP E
```

β) Αρχιτεκτονική που βασίζεται στη χρήση συσσωρευτή:

```
LOAD B
ADD C
MUL A
STORE E
LOAD D
ADD E
STORE E
```

γ) Αρχιτεκτονική καταχωρητή-μνήμης:

```
LOAD R1, A
MUL R1, D
ADD R1, C
STORE E, R1
LOAD R2, B
ADD R2, A
ADD R2, E
STORE E, R2
```


δ) Αρχιτεκτονική καταχωρητή-καταχωρητή:

```

LOAD R1, A
LOAD R2, B
LOAD R3, C
ADD R4, R2, R3      /  $R4 \leftarrow R2 + R3$ 
SUB R4, R4, R1      /  $R4 \leftarrow R4 - R1$ 
LOAD R5, D
MUL R5, R4, R5      /  $R5 \leftarrow R4 \times R5$ 
DIV R5, R5, R1      /  $R5 \leftarrow R5 / R1$ 
STORE E, R5
    
```

Απάντηση

α) Αρχιτεκτονική που βασίζεται στη χρήση μηχανισμού στοίβας:

ΠΡΟΓΡΑΜΜΑ	ΣΤΟΙΒΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΝΤΟΛΗΣ (ΚΟΡΥΦΗ ΣΤΑ ΑΡΙΣΤΕΡΑ)
PUSH A	A
PUSH B	B, A
PUSH C	C, B, A
ADD	$(C + B), A$
PUSH D	D, $(C + B), A$
ADD	$(D + (C + B)), A$
MUL	$((D + C + B) \times A)$
PUSH A	A, $((D + C + B) \times A)$
ADD	$(A + ((D + C + B) \times A))$
POP E	σκουπίδια

$$E = A + (D + C + B) \times A$$

β) Αρχιτεκτονική συσσωρευτή:

ΠΡΟΓΡΑΜΜΑ	ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ ΣΥΣΣΩΡΕΥΤΗ Σ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΝΤΟΛΗΣ
LOAD B	$\Sigma = B$
ADD C	$\Sigma = (B + C)$
MUL A	$\Sigma = (B + C) \times A$
STORE E	$E = (B + C) \times A$
LOAD D	$\Sigma = D$
ADD E	$\Sigma = D + E = D + (B + C) \times A$
STORE E	$E = D + (B + C) \times A$

$$E = D + (B + C) \times A$$

γ) Αρχιτεκτονική καταχωρητή-μνήμης:

ΠΡΟΓΡΑΜΜΑ	ΑΠΟΤΕΛΕΣΜΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΝΤΟΛΗΣ
LOAD R1, A	$R1 = A$
MUL R1, D	$R1 = R1 \times D = A \times D$
ADD R1, C	$R1 = R1 + C = (A \times D) + C$
STORE E, R1	$E = R1 = (A \times D) + C$
LOAD R2, B	$R2 = B$
ADD R2, A	$R2 = R2 + A = B + A$
ADD R2, E	$R2 = R2 + E = B + A + ((A \times D) + C)$
STORE E, R2	$E = B + A + (A \times D) + C$

$$E = B + A + (A \times D) + C$$

δ) Αρχιτεκτονική καταχωρητή-καταχωρητή:

ΠΡΟΓΡΑΜΜΑ	ΑΠΟΤΕΛΕΣΜΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΝΤΟΛΗΣ
LOAD R1, A	$R1 = A$
LOAD R2, B	$R2 = B$
LOAD R3, C	$R3 = C$
ADD R4, R2, R3	$R4 = R2 + R3 = B + C$
SUB R4, R4, R1	$R4 = R4 - R1 = (B + C) - A$
LOAD R5, D	$R5 = D$
MUL R5, R4, R5	$R5 = R4 \times R5 = ((B + C) - A) \times D$
DIV R5, R5, R1	$R5 = R5 / R1 = ((B + C - A) \times D) / A$
STORE E, R5	$E = R5 = ((B + C - A) \times D) / A$

$$E = ((B + C - A) \times D) / A$$

Άσκηση 2.12

A. Για τον υπολογισμό της έκφρασης “ $Y=A+B \times C+B \times E$ ” να γραφεί πρόγραμμα για υπολογιστή που βασίζεται στη χρήση

- i. του μηχανισμού στοίβας
- ii. του συσσωρευτή
- iii. 16 καταχωρητών γενικού σκοπού με εντολές iii.1 μόνο καταχωρητή-κατάχωρητή (ή φόρτωσης-αποθήκευσης) με δύο τελούμενα και iii.2 καταχωρητή-καταχωρητή και καταχωρητή-μνήμης με δύο τελούμενα

B. Να υπολογιστεί ο χώρος που καταλαμβάνει κάθε πρόγραμμα εάν ο κωδικός λειτουργίας κάθε εντολής καταλαμβάνει μία ψηφιολέξη (byte), ο τρόπος διευθυνσιοδότησης του συστήματος μνήμης που χρησιμοποιείται στις εντολές είναι ο κατ’ ευθείαν τρόπος (direct addressing mode), οι διευθύνσεις είναι των 32 δυαδικών ψηφίων και οι μηχανές που βασίζονται στη χρήση καταχωρητών γενικού σκοπού έχουν 16 καταχωρητές γενικού σκοπού. Κάθε εντολή καταλαμβάνει ακέραιο αριθμό ψηφιολέξεων.

Γ. Θεωρήστε ότι τα δεδομένα είναι μιας ψηφιολέξης, η αρτηρία δεδομένων μεταξύ της Κεντρικής Μονάδας Επεξεργασίας (ΚΜΕ) και του συστήματος μνήμης έχει εύρος μιας ψηφιολέξης (byte) και σε κάθε διεύθυνση του συστήματος μνήμης αντιστοιχεί μια ψηφιολέξη. Στην περίπτωση μηχανής που βασίζεται στο μηχανισμό στοίβας, η στοίβα υλοποιείται στην ΚΜΕ. Πόσες προσπελάσεις στη μνήμη απαιτεί το κάθε πρόγραμμα;

Απάντηση

A. i. Υπολογιστής που βασίζεται στη χρήση του μηχανισμού στοίβας.

Στην περίπτωση μηχανής που βασίζεται στη χρήση του μηχανισμού στοίβας, για να γράψουμε εύκολα ένα πρόγραμμα για ένα αριθμητικό υπολογισμό, θα πρέπει καταρχήν να μετατρέψουμε την αριθμητική έκφραση από “infix notation” σε “postfix notation”. Μία αριθμητική έκφραση είναι σε “infix notation” όταν ο τελεστής βρίσκεται μεταξύ των δεδομένων που χρησιμοποιούνται στην πράξη, πχ $A \times B$. Μία αριθμητική έκφραση είναι σε “postfix notation” όταν ο τελεστής βρίσκεται μετά τα δεδομένα που χρησιμοποιούνται στην πράξη, πχ $AB \times$. Λαμβάνοντας υπόψη τις προτεραιότητες των πράξεων και τις παρενθέσεις, μετατρέπουμε καταρχήν τις πράξεις μεγαλύτερης προτεραιότητας από “infix notation” σε “postfix notation”. Κάθε πράξη που έχει μετατραπεί σε “postfix notation” στη συνέχεια θεωρείται ως δεδομένο.

Στις παρακάτω πράξεις κάθε υπογραμμισμένη έκφραση είναι σε “postfix notation”.

$$Y = A + B \times C + B \times E \rightarrow Y = A + \underline{BC \times} + \underline{BE \times} \rightarrow Y = \underline{ABC \times} + \underline{BE \times} \rightarrow Y = \underline{ABC \times} + \underline{BE \times} +$$

Τώρα διατρέχοντας τη σχέση, την οποία έχουμε σε “postfix notation”, από τα αριστερά προς τα δεξιά, κάθε φορά που θα συναντάμε τελούμενο θα γράφουμε την εντολή Push, κάθε φορά που θα συναντάμε τελεστή θα γράφουμε την εντολή που αντιστοιχεί στον τελεστή και όταν συναντήσουμε το σύμβολο της ισότητας θα γράψουμε την εντολή POP. Επομένως το πρόγραμμα είναι:

ΠΡΟΓΡΑΜΜΑ	ΣΤΟΙΒΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΝΤΟΛΗΣ (ΚΟΡΥΦΗ ΣΤΑ ΑΡΙΣΤΕΡΑ)
PUSH A	A
PUSH B	B, A
PUSH C	C, B, A
MUL	$B \times C$, A
ADD	$B \times C + A$
PUSH B	B, $B \times C + A$
PUSH E	E, B, $B \times C + A$
MUL	$E \times B$, $B \times C + A$
ADD	$E \times B + B \times C + A$
POP Y	$Y = E \times B + B \times C + A$

B. i.

Επειδή στις εντολές χρησιμοποιείται ο κατ' ευθείαν τρόπος διευθυνσιοδότησης του συστήματος μνήμης και οι διευθύνσεις είναι των 32 δυαδικών ψηφίων, για κάθε δεδομένο που εμφανίζεται στην εντολή της συμβολικής γλώσσας, στην εντολή σε επίπεδο γλώσσας μηχανής θα υπάρχει ένα πεδίο 4 ψηφιολέξεων. Επίσης σε κάθε εντολή υπάρχει ένα πεδίο μιας ψηφιολέξης για τον κωδικό λειτουργίας. Επομένως,

Μία εντολή «PUSH X» ή «POP X» καταλαμβάνει 1 ψηφιολέξη για τον κωδικό λειτουργίας της και 4 ψηφιολέξεις για τη διεύθυνση του X, δηλαδή συνολικά 5 ψηφιολέξεις.

Μία εντολή «MUL» ή «ADD» καταλαμβάνει 1 ψηφιολέξη για τον κωδικό λειτουργίας της.

Επομένως το πρόγραμμα καταλαμβάνει $6 \times 5 + 4$ ψηφιολέξεις = 34 ψηφιολέξεις της κύριας μνήμης.

Γ. i.

Επειδή τα δεδομένα είναι μιας ψηφιολέξης και η αρτηρία δεδομένων μεταξύ της Κεντρικής Μονάδας Επεξεργασίας (ΚΜΕ) και του συστήματος μνήμης έχει εύρος μιας ψηφιολέξης (byte) και σε κάθε διεύθυνση του συστήματος μνήμης αντιστοιχεί μια ψηφιολέξη έχουμε:

Για την εκτέλεση μιας εντολής «PUSH X» απαιτούνται 5 προσπελάσεις για την προσκόμιση της εντολής και 1 προσπέλαση για την προσκόμιση του δεδομένου.

Για την εκτέλεση μιας εντολής «POP X» απαιτούνται 5 προσπελάσεις για την προσκόμιση της εντολής και 1 προσπέλαση για την αποθήκευση του δεδομένου.

Για την εκτέλεση μιας εντολής «ADD» ή «MUL» απαιτείται η προσπέλαση μόνο της εντολής, δηλαδή 1 προσπέλαση του συστήματος μνήμης.

Επομένως για την εκτέλεση του προγράμματος απαιτούνται $6 \times 6 + 4 = 40$ προσπελάσεις του συστήματος μνήμης.

A. ii. Υπολογιστής που βασίζεται στη χρήση συσσωρευτή.

Υπενθυμίζουμε ότι η έκφραση είναι: $Y = A + B \times C + B \times E$

ΠΡΟΓΡΑΜΜΑ	ΠΕΡΙΕΧΟΜΕΝΟ ΤΟΥ ΣΥΣΣΩΡΕΥΤΗ Σ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΝΤΟΛΗΣ
LOAD B	$\Sigma = B$
MUL C	$\Sigma = \Sigma \times C = B \times C$
ADD A	$\Sigma = \Sigma + A = B \times C + A$
STORE X	$X = B \times C + A$
LOAD B	$\Sigma = B$
MUL E	$\Sigma = \Sigma \times E = B \times E$
ADD X	$\Sigma = \Sigma + X = B \times E + X = B \times E + B \times C + A$
STORE Y	$Y = B \times E + B \times C + A$

B. ii.

Σ' αυτή την περίπτωση οποιαδήποτε από τις εντολές που χρησιμοποιήθηκαν καταλαμβάνει 1 ψηφιολέξη για τον κωδικό λειτουργίας και 4 ψηφιολέξεις για τη διεύθυνση του δεδομένου, δηλαδή 5 ψηφιολέξεις συνολικά. Επομένως το πρόγραμμα καταλαμβάνει 8×5 ψηφιολέξεις = 40 ψηφιολέξεις

Γ. ii.

Σ' αυτή την περίπτωση για την εκτέλεση οποιασδήποτε από τις εντολές που χρησιμοποιήθηκαν απαιτούνται 5 προσπελάσεις της μνήμης για την προεπέλαση της εντολής και 1 προσπέλαση για την προσπέλαση των δεδομένων, δηλαδή συνολικά 6 προσπελάσεις. Επομένως για την εκτέλεση του προγράμματος απαιτούνται 8×6 προσπελάσεις = 48 προσπελάσεις του συστήματος μνήμης.

A.iii.1 Υπολογιστής που βασίζεται στη χρήση καταχωρητών γενικού σκοπού και έχει εντολές μόνο καταχωρητή-καταχωρητή με δύο τελούμενα.

$$Y = A + B \times C + B \times E$$

ΠΡΟΓΡΑΜΜΑ	ΑΠΟΤΕΛΕΣΜΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΝΤΟΛΗΣ
LOAD R1, B	$R1=B$
LOAD R2, C	$R2=C$
MUL R2, R1	$R2=R2 \times R1 = C \times B$
LOAD R3, E	$R3=E$
MUL R1, R3	$R1=R1 \times R3 = B \times E$
ADD R1, R2	$R1=R1+R2 = B \times E + C \times B$
LOAD R2, A	$R2=A$
ADD R1, R2	$R1=R1+R2 = B \times E + C \times B + A$
STORE Y, R1	$Y=R1 = B \times E + C \times B + A$

B.iii.1

Μία εντολή «LOAD R, X» ή «STORE X, R» καταλαμβάνει 1 ψηφιολέξη για τον κωδικό λειτουργίας, 4 δυαδικά ψηφία για τη δήλωση του καταχωρητή R (ο υπολογιστής μας έχει 16 καταχωρητές) και 4 ψηφιολέξεις για τη διεύθυνση του δεδομένου X στη μνήμη. Επειδή κάθε εντολή καταλαμβάνει ακέραιο αριθμό ψηφιολέξεων, οποιαδήποτε από αυτές τις εντολές θα καταλαμβάνει 6 ψηφιολέξεις.

Μία εντολή «ADD R1, R2» ή «MUL R1, R2» καταλαμβάνει 1 ψηφιολέξη για τον κωδικό λειτουργίας, 4 δυαδικά ψηφία για τη δήλωση του καταχωρητή R1 και 4 δυαδικά ψηφία για τη δήλωση του καταχωρητή R2, δηλαδή συνολικά 2 ψηφιολέξεις.

Επομένως το πρόγραμμα καταλαμβάνει $5 \times 6 + 4 \times 2$ ψηφιολέξεις = 38 ψηφιολέξεις.

Γ.iii.1

Για την εκτέλεση μιας εντολής «LOAD R, X» απαιτούνται 6 προσπελάσεις για την προσκόμιση της εντολής και 1 προσπέλαση για την προσκόμιση του δεδομένου, δηλαδή συνολικά 7 προσπελάσεις.

Για την εκτέλεση μιας εντολής «STORE Y, R» απαιτούνται 6 προσπελάσεις για την προσκόμιση της εντολής και 1 προσπέλαση για την αποθήκευση του δεδομένου, δηλαδή συνολικά 7 προσπελάσεις.

Για την εκτέλεση μιας εντολής «ADD R1, R2» ή «MUL R1, R2» απαιτούνται μόνο 2 προσπελάσεις για την προσκόμιση της εντολής.

Επομένως για την εκτέλεση του προγράμματος απαιτούνται $5 \times 7 + 4 \times 2$ προσπελάσεις = 43 προσπελάσεις του συστήματος μνήμης.

A. iii.2 Υπολογιστής που βασίζεται στη χρήση καταχωρητών γενικού σκοπού και έχει εντολές καταχωρητή-καταχωρητή και καταχωρητή-μνήμης με δύο τελούμενα.

ΠΡΟΓΡΑΜΜΑ	ΑΠΟΤΕΛΕΣΜΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΝΤΟΛΗΣ
LOAD R1, B	$R1=B$
MUL R1, C	$R1=R1 \times C=B \times C$
LOAD R2, B	$R2=B$
MUL R2, E	$R2=R2 \times E=B \times E$
ADD R1, R2	$R1=R1+R2=C \times B+B \times E$
ADD R1, A	$R1=R1+A=C \times B+B \times E+A$
STORE Y, R1	$Y=R1=C \times B+B \times E+A$

$$Y=A+B \times C+B \times E$$

B. iii.2

Οποιαδήποτε από τις εντολές «LOAD R, X» ή «STORE X, R» ή «MUL R, X» ή «ADD R, X» καταλαμβάνει 1 ψηφιολέξη για τον κωδικό λειτουργίας, 4 δυαδικά ψηφία για τη δήλωση του καταχωρητή R και 4 ψηφιολέξεις για τη δήλωση της διεύθυνσης X. Επειδή κάθε εντολή καταλαμβάνει ακέραιο αριθμό ψηφιολέξεων, κάθε μια από τις παραπάνω εντολές θα καταλαμβάνει 6 ψηφιολέξεις.

Η εντολή «ADD R1, R2» καταλαμβάνει 1 ψηφιολέξη για τον κωδικό λειτουργίας, 4 δυαδικά ψηφία για τη δήλωση του καταχωρητή R1 και 4 δυαδικά ψηφία για τη δήλωση του καταχωρητή R2, δηλαδή συνολικά 2 ψηφιολέξεις.

Επομένως το πρόγραμμα καταλαμβάνει $6 \times 6 + 2$ ψηφιολέξεις = 38 ψηφιολέξεις.

Γ. iii.2

Για την εκτέλεση μιας εντολής «LOAD R, X» ή «ADD R1, X» ή «MUL R1, X» απαιτούνται 6 προσπελάσεις της μνήμης για την προσκόμιση της εντολής και 1 προσπέλαση για την προσκόμιση του δεδομένου, δηλαδή συνολικά 7 προσπελάσεις.

Για την εκτέλεση μιας εντολής «STORE Y, R» απαιτούνται 6 προσπελάσεις για την προσκόμιση της εντολής και 1 προσπέλαση για την αποθήκευση του δεδομένου, δηλαδή συνολικά 7 προσπελάσεις.

Για την εκτέλεση μιας εντολής «ADD R1, R2» απαιτούνται μόνο 2 προσπελάσεις για την προσκόμιση της εντολής.

Επομένως για την εκτέλεση του προγράμματος απαιτούνται $6 \times 7 + 2$ προσπελάσεις = 44 προσπελάσεις του συστήματος μνήμης.

Άσκηση 2.13 (2.8 βιβλίου)

Θεωρήστε υπολογιστή απλού συνόλου εντολών στον οποίο η αρτηρία διευθύνσεων είναι των 32 δυαδικών ψηφίων, έχει 32 καταχωρητές γενικού σκοπού και η κύρια μνήμη έχει μια ψηφιολέξη ανά θέση μνήμης. Όλες οι εντολές έχουν μήκος 32 δυαδικά ψηφία και ο κωδικός λειτουργίας καταλαμβάνει τα 8 δυαδικά ψηφία της εντολής. Σχολιάστε την υλοποίηση των ακόλουθων δύο εντολών:

α. BRE r1, r2, d // εάν $r1-r2=0$ τότε $ΜΠ=ΜΠ+d$

β. BRE d // εάν το αποτέλεσμα της προηγούμενης αριθμητικής ή λογικής πράξης ήταν ίσο με το μηδέν τότε $ΜΠ=ΜΠ+d$.

ΜΠ είναι ο Μετρητής Προγράμματος.

Απάντηση

α. BRE r1, r2, d // εάν $r1-r2=0$ τότε $ΜΠ=ΜΠ+d$

Αφού ο υπολογιστής έχει 32 καταχωρητές, συνεπάγεται ότι για να αναφερθούμε σ' ένα καταχωρητή χρειαζόμαστε ένα δυαδικό αριθμό των 5 δυαδικών ψηφίων. Επομένως η εντολή σ' αυτή την περίπτωση μπορεί να έχει την ακόλουθη μορφή:

ΠΣΨ										ΛΣΨ	
31	-	24	23	-	19	18	-	14	13	-	0
ΚΛ			r1			r2			d		

Κωδικός λειτουργίας, ΚΛ, 8 δυαδικών ψηφίων	Αριθμός αναφοράς καταχωρητή, 5 δυαδικά ψηφία	Αριθμός αναφοράς καταχωρητή, 5 δυαδικά ψηφία	Αριθμός μετατόπισης (displacement)
-----------------------------------------------------	----------------------------------------------------------	----------------------------------------------------------	---------------------------------------

Λαμβάνοντας υπόψη ότι ο αριθμός μετατόπισης είναι ένας αριθμός των 14 δυαδικών ψηφίων σε παράσταση συμπληρώματος ως προς 2, συμπεραίνουμε ότι το άλμα μπορεί να γίνει μέσα στην περιοχή των φυσικών διευθύνσεων από $ΜΠ-2^{13}$ έως $ΜΠ+2^{13}-1$.

β. BRE d // εάν το αποτέλεσμα της προηγούμενης αριθμητικής ή λογικής πράξης ήταν ίσο με το μηδέν τότε $ΜΠ=ΜΠ+d$.

Σ' αυτή την περίπτωση στην εντολή υπάρχει μόνο το πεδίο του κωδικού λειτουργίας και ο αριθμός μετατόπισης, όπως φαίνεται στο επόμενο σχήμα. Επομένως το πλήθος των δυαδικών ψηφίων που καταλαμβάνει ο αριθμός μετατόπισης είναι μεγαλύτερο και το άλμα μπορεί να γίνει μέσα στην περιοχή των φυσικών διευθύνσεων από $ΜΠ-2^{23}$ έως $ΜΠ+2^{23}-1$.

ΠΣΨ				ΛΣΨ			
31	-	24	23	-			0
ΚΛ				d			

Κωδικός
λειτουργίας,
ΚΛ, 8 δυαδικών
ψηφίων

Αριθμός μετατόπισης (displacement)

Άσκηση 2.14 (2.9 βιβλίου)

Θεωρήστε ότι σ' ένα επεξεργαστή δεν υποστηρίζονται οι παρακάτω εντολές:

1. ADD r1, r2 // $r1 \leftarrow r1 + r2$
2. ADD r1, #m // $r1 \leftarrow r1 + m$
3. ADD r1, m(r2) // $r1 \leftarrow r1 + \text{Mem}(m + r2)$
4. ADD r1, (r2) // $r1 \leftarrow r1 + \text{Mem}(r2)$
5. ADD r1, (r2+r3) // $r1 \leftarrow r1 + \text{Mem}(r2 + r3)$
6. ADD r1, @(r2) // $r1 \leftarrow r1 + \text{Mem}(\text{Mem}(r2))$
7. ADD r1, (r2)+ // $r1 \leftarrow r1 + \text{Mem}(r2)$ και $r2 \leftarrow r2 + d$, το d είναι μια προκαθορισμένη τιμή
8. ADD r1, -(r2) // $r2 \leftarrow r2 - d$ και $r1 \leftarrow r1 + \text{Mem}(r2)$, το d είναι μια προκαθορισμένη τιμή

α. Να περιγράψετε τον τρόπο διευθυνσιοδότησης που χρησιμοποιείται στις παραπάνω εντολές για την προσπέλαση κάθε τελούμενου.

β. Να υλοποιήσετε τις παραπάνω εντολές χρησιμοποιώντας συνδυασμούς από τις κάτωθι εντολές που διαθέτει ο υπολογιστής.

LOAD r1, (r2)	// $r1 \leftarrow \text{Mem}(r2)$
ADD r1, r2, #m	// $r1 \leftarrow r1 + r2 + m, m \geq 0$
CLEAR r1	// $r1 = 0$
COMP r1	// $r1 = \text{το συμπλήρωμα του } r1 \text{ ως προς ένα.}$

Απάντηση

Θα απαντήσουμε τα ερωτήματα α και β για κάθε μία εντολή.

1. ADD r1, r2 // $r1 \leftarrow r1 + r2$

α. Για την προσπέλαση και των δύο δεδομένων που θα χρησιμοποιηθούν για την εκτέλεση της εντολής, καθώς και για την αποθήκευση του αποτελέσματος, χρησιμοποιείται ο κατ' ευθείαν τρόπος διευθυνσιοδότησης καταχωρητή.

β. ADD r1, r2, #0 // $r1 \leftarrow r1 + r2 + 0$

2. ADD r1, #m // $r1 \leftarrow r1 + m$

α. Για την ανάγνωση του περιεχομένου του καταχωρητή r1 και την αποθήκευση του αποτελέσματος στον ίδιο καταχωρητή χρησιμοποιείται ο κατ' ευθείαν τρόπος διευθυνσιοδότησης καταχωρητή. Για την προσπέλαση της σταθεράς m χρησιμοποιείται ο άμεσος τρόπος διευθυνσιοδότησης.

β. CLEAR r2 // $r2 = 0$
ADD r1, r2, #m // $r1 \leftarrow r1 + r2 + m, m \geq 0$

3. ADD r1, m(r2) // $r1 \leftarrow r1 + \text{Mem}(m + r2)$

α. Για την ανάγνωση του περιεχομένου του καταχωρητή r1 και την αποθήκευση του αποτελέσματος στον ίδιο καταχωρητή χρησιμοποιείται ο κατ' ευθείαν τρόπος διευθυνσιοδότησης καταχωρητή. Για την προσπέλαση της μνήμης χρησιμοποιείται ο σχετικός τρόπος διευθυνσιοδότησης σύμφωνα με τον οποίο η διεύθυνση της μνήμης που θα προσπελαστεί καθορίζεται από το άθροισμα του περιεχομένου του καταχωρητή r2 και της σταθεράς m που δηλώνονται στην εντολή.

β. Το περιεχόμενο του r2 δεν πρέπει να αλλάξει.

CLEAR r3 // $r3 = 0$
ADD r3, r2, #m // $r3 \leftarrow 0 + r2 + m$
LOAD r3, (r3) // $r3 \leftarrow \text{Mem}(r3) = \text{Mem}(r2 + m)$
ADD r1, r3, #0 // $r1 \leftarrow r1 + r3 + 0$

4. ADD r1, (r2) // $r1 \leftarrow r1 + \text{Mem}(r2)$

α. Για την ανάγνωση του περιεχομένου του καταχωρητή r1 και την αποθήκευση του αποτελέσματος στον ίδιο καταχωρητή χρησιμοποιείται ο κατ' ευθείαν τρόπος διευθυνσιοδότησης καταχωρητή. Για την προσπέλαση της μνήμης χρησιμοποιείται έμμεσος τρόπος διευθυνσιοδότησης με χρήση καταχωρητή.

β. Το περιεχόμενο του r2 δεν πρέπει να αλλάξει.

```
LOAD r3, (r2)           // r3 ← Mem(r2)
ADD r1, r3, #0          // r1 ← r1 + r3 + 0
```

5. ADD r1, (r2+r3) // r1 ← r1 + Mem(r2+r3)

α. Για την ανάγνωση του περιεχομένου του καταχωρητή r1 και την αποθήκευση του αποτελέσματος στον ίδιο καταχωρητή χρησιμοποιείται ο κατ' ευθείαν τρόπος διευθυνσιοδότησης καταχωρητή. Για την προσπέλαση της μνήμης χρησιμοποιείται ένα είδος έμμεσου τρόπου διευθυνσιοδότησης με χρήση καταχωρητή σύμφωνα με τον οποίο η διεύθυνση της μνήμης που θα προσπελαστεί καθορίζεται από το άθροισμα των περιεχομένων των καταχωρητών r2 και r3 που δηλώνονται στην εντολή. Ο τρόπος προσπέλασης της μνήμης θα μπορούσε να χαρακτηριστεί και ως σχετικός τρόπος διευθυνσιοδότησης διότι η διεύθυνση της θέσης μνήμης που θα προσπελαστεί σχηματίζεται από το άθροισμα των περιεχομένων δύο καταχωρητών.

β. Το περιεχόμενο των r2 και r3 δεν πρέπει να αλλάξει.

```
CLEAR r4                // r4 = 0
ADD r4, r2, #0           // r4 ← 0 + r2
ADD r4, r3, #0           // r4 ← r4 + r3 + 0 = r2 + r3
LOAD r4, (r4)            // r4 ← Mem(r4)
ADD r1, r4, #0           // r1 ← r1 + r4 + 0
```

6. ADD r1, @(r2) // r1 ← r1 + Mem(Mem(r2))

α. Για την ανάγνωση του περιεχομένου του καταχωρητή r1 και την αποθήκευση του αποτελέσματος στον ίδιο καταχωρητή, χρησιμοποιείται ο κατ' ευθείαν τρόπος διευθυνσιοδότησης καταχωρητή. Για την προσπέλαση της μνήμης χρησιμοποιείται έμμεσος τρόπος διευθυνσιοδότησης δύο επιπέδων. Στο πρώτο επίπεδο έχουμε έμμεσο τρόπο διευθυνσιοδότησης με χρήση καταχωρητή, του καταχωρητή r2. Στο δεύτερο επίπεδο έχουμε έμμεσο τρόπο διευθυνσιοδότησης με χρήση της κύριας μνήμης, συγκεκριμένα της θέσης μνήμης η διεύθυνση της οποίας περιέχεται στον καταχωρητή r2.

β. Το περιεχόμενο του r2 δεν πρέπει να αλλάξει.

```
LOAD r3, (r2)           // r3 ← Mem(r2)
LOAD r3, (r3)           // r3 ← Mem(r3)
ADD r1, r3, #0          // r1 ← r1 + r3 + 0
```


7. ADD r1, (r2)+ // $r1 \leftarrow r1 + \text{Mem}(r2)$ και $r2 \leftarrow r2 + d$, το d είναι
μια προκαθορισμένη τιμή

α. Για την ανάγνωση του περιεχομένου του καταχωρητή $r1$ και την αποθήκευση του αποτελέσματος στον ίδιο καταχωρητή χρησιμοποιείται ο κατ' ευθείαν τρόπος διευθυνσιοδότησης καταχωρητή. Για την προσπέλαση της μνήμης χρησιμοποιείται έμμεσος τρόπος διευθυνσιοδότησης με χρήση καταχωρητή. Μετά τη χρησιμοποίηση του περιεχομένου του καταχωρητή, αυτό αυξάνεται κατά d .

β. `LOAD r3, (r2)` // $r3 \leftarrow \text{Mem}(r2)$
`CLEAR r4` // $r4 = 0$
`ADD r2, r4, #d` // $r2 \leftarrow r2 + 0 + d$
`ADD r1, r3, #0` // $r1 \leftarrow r1 + r3 + 0$

8. ADD r1, -(r2) // $r2 \leftarrow r2 - d$ και $r1 \leftarrow r1 + \text{Mem}(r2)$, το d είναι μια
προκαθορισμένη τιμή

α. Για την ανάγνωση του περιεχομένου του καταχωρητή $r1$ και την αποθήκευση του αποτελέσματος στον ίδιο καταχωρητή χρησιμοποιείται ο κατ' ευθείαν τρόπος διευθυνσιοδότησης καταχωρητή. Για την προσπέλαση της μνήμης χρησιμοποιείται έμμεσος τρόπος διευθυνσιοδότησης με χρήση καταχωρητή. Πριν τη χρησιμοποίηση του περιεχομένου του καταχωρητή $r2$ για την προσπέλαση της μνήμης, αυτό ελαττώνεται κατά d .

β. `CLEAR r3` // $r3 = 0$
`ADD r3, r3, #d` // $r3 \leftarrow 0 + 0 + d = d$
`COMP r3` // $r3 =$ το συμπλήρωμα του d ως προς ένα.
`CLEAR r4` // $r4 = 0$
`ADD r3, r4, #1` // $r3 \leftarrow$ το συμπλήρωμα του d ως προς ένα $+ 0 + 1 =$
το συμπλήρωμα του d ως προς δύο
`ADD r2, r3, #0` // $r2 \leftarrow r2 - d$
`LOAD r3, (r2)` // $r3 \leftarrow \text{Mem}(r2)$
`ADD r1, r3, #0` // $r1 \leftarrow r1 + r3 + 0$

ΚΕΦΑΛΑΙΟ 3

Ασκήσεις για την Κεντρική Μονάδα Επεξεργασίας

Άσκηση 3.1 (3.1 βιβλίου)

Οι αριθμοί που ακολουθούν είναι σε αναπαράσταση συμπληρώματος ως προς 2.

α. 10000110

β. 11000000

γ. 01111100

δ. 01000001

Να εκτελεστούν οι πράξεις $\alpha+\beta$, $\gamma+\delta$, $\alpha+\gamma$ και $\beta+\delta$ σε κάθε μία από τις επόμενες περιπτώσεις, να δοθεί το αποτέλεσμα σε δυαδική και δεκαδική μορφή και να σχολιαστεί αν το αποτέλεσμα είναι σωστό ή όχι και γιατί:

1. Για την πράξη χρησιμοποιείται αθροιστής των 8 δυαδικών ψηφίων
2. Για την πράξη χρησιμοποιείται αθροιστής των 16 δυαδικών ψηφίων

Απάντηση

Επειδή το πιο σημαντικό δυαδικό ψηφίο των α και β έχει την τιμή 1, οι αριθμοί αυτοί είναι αρνητικοί και για να πάρουμε το μέγεθός τους θα πρέπει να πάρουμε το συμπλήρωμά τους ως προς 2.

$$\begin{array}{r} \alpha \qquad \qquad \qquad 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0 \\ \text{Συμπλήρωμα ως προς 1} \quad 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1 \\ \qquad \qquad \qquad + \qquad \qquad \qquad 1 \\ \text{Συμπλήρωμα ως προς 2} \quad 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0 \end{array}$$

Επομένως $\alpha = -122_{(10)}$

$$\begin{array}{r} \beta \qquad \qquad \qquad 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0 \\ \text{Συμπλήρωμα ως προς 1} \quad 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1 \\ \qquad \qquad \qquad + \qquad \qquad \qquad 1 \\ \text{Συμπλήρωμα ως προς 2} \quad 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0 \end{array}$$

$$\beta = -64_{(10)}$$

Επειδή το πιο σημαντικό δυαδικό ψηφίο των γ και δ έχει την τιμή 0, οι αριθμοί αυτοί είναι θετικοί, επομένως το μέγεθός τους δίνεται από όλα τα δυαδικά ψηφία.


$$\gamma = 01111100 = 124_{(10)}$$

$$\delta = 01000001 = 65_{(10)}$$

1. Αφού οι αριθμοί είναι των 8 δυαδικών ψηφίων και ο αθροιστής είναι των 8 δυαδικών ψηφίων θα εκτελέσουμε κατ' ευθείαν την πράξη. Υπενθυμίζουμε ότι στην πρόσθεση μεταξύ αριθμών σε αναπαράσταση συμπληρώματος ως προς 2 το κρατούμενο εξόδου το αγνοούμε. Η υπερχείλιση σ' αυτή την περίπτωση δίνεται από τη σχέση $Y = \kappa_8 \oplus \kappa_7$, όπου κ_7 είναι το κρατούμενο εισόδου στην τελευταία, πιο σημαντική, βαθμίδα του αθροιστή και κ_8 είναι το κρατούμενο εξόδου.

$\alpha + \beta$

$$\begin{array}{r}
 \alpha \quad 10000110 \\
 \beta \quad + 11000000 \\
 \hline
 \text{απ} \quad \textcircled{1}01000110
 \end{array}$$


 το κρατούμενο εξόδου το αγνοούμε

$Y = \kappa_8 \oplus \kappa_7 = 1 \oplus 0 = 1$ επομένως έχουμε υπερχείλιση.

(Παρατηρούμε ότι το πιο σημαντικό δυαδικό ψηφίο του αποτελέσματος είναι ίσο με 0, άρα προσθέτοντας δύο αρνητικούς αριθμούς πήραμε ένα αριθμό που είναι θετικός. Αυτό δεν είναι δυνατόν να συμβεί, άρα έχουμε υπερχείλιση.)

Πράγματι $\alpha + \beta = -122 - 64 = -186$, ο οποίος είναι μικρότερος του αριθμού $-2^7 = -128$ που είναι ο μικρότερος αριθμός που μπορεί να παρασταθεί με 8 δυαδικά ψηφία σε αναπαράσταση συμπληρώματος ως προς 2.

$\gamma + \delta$

$$\begin{array}{r}
 \gamma \quad 01111100 \\
 \delta \quad + 01000001 \\
 \hline
 \text{απ} \quad 10111101
 \end{array}$$


$Y = \kappa_8 \oplus \kappa_7 = 0 \oplus 1 = 1$ επομένως έχουμε υπερχείλιση.

(Παρατηρούμε ότι το πιο σημαντικό δυαδικό ψηφίο του αποτελέσματος είναι ίσο με 1, άρα προσθέτοντας δύο θετικούς αριθμούς πήραμε ένα αριθμό που είναι αρνητικός. Αυτό δεν είναι δυνατόν να συμβεί, άρα έχουμε υπερχείλιση.)

Πράγματι $\gamma + \delta = 124 + 65 = 189$, ο οποίος είναι μεγαλύτερος του αριθμού $2^7 - 1 = 127$ που είναι ο μεγαλύτερος αριθμός που μπορεί να παρασταθεί με 8 δυαδικά ψηφία σε αναπαράσταση συμπληρώματος ως προς 2.

$\alpha + \gamma$

$$\begin{array}{r}
 \alpha \quad 10000110 \\
 \gamma \quad + 01111100 \\
 \hline
 \alpha\pi \quad \textcircled{1}00000010
 \end{array}$$



 το κρατούμενο εξόδου το αγνοούμε

$Y = \kappa_8 \oplus \kappa_7 = 1 \oplus 1 = 0$ επομένως δεν έχουμε υπερχείλιση.

Πράγματι $\alpha + \gamma = -122_{(10)} + 124_{(10)} = 2_{(10)}$, ο οποίος ισούται με το αποτέλεσμα που λάβαμε προσθέτοντας τους δυαδικούς αριθμούς σε αναπαράσταση συμπληρώματος ως προς 2.

$\beta + \delta$

$$\begin{array}{r}
 \beta \quad 11000000 \\
 \delta \quad + 01000001 \\
 \hline
 \alpha\pi \quad \textcircled{1}00000001
 \end{array}$$


 το κρατούμενο εξόδου το αγνοούμε


$Y = \kappa_8 \oplus \kappa_7 = 1 \oplus 1 = 0$ επομένως δεν έχουμε υπερχείλιση.

Πράγματι $\beta + \delta = -64_{(10)} + 65_{(10)} = 1_{(10)}$ ο οποίος ισούται με το αποτέλεσμα που λάβαμε προσθέτοντας τους δυαδικούς αριθμούς σε αναπαράσταση συμπληρώματος ως προς 2.

2. Αφού οι αριθμοί είναι των 8 δυαδικών ψηφίων και ο αθροιστής είναι των 16 δυαδικών ψηφίων θα πρέπει να κάνουμε επέκταση προσήμου. Υπενθυμίζουμε ότι για να κάνουμε επέκταση προσήμου επαναλαμβάνουμε το πιο σημαντικό δυαδικό ψηφίο του αριθμού όσες φορές χρειάζεται, στη συγκεκριμένη περίπτωση 8 φορές. Η υπερχείλιση σ' αυτή την περίπτωση δίνεται από τη σχέση $Y = \kappa_{16} \oplus \kappa_{15}$, όπου κ_{15} είναι το κρατούμενο εισόδου στην τελευταία, πιο σημαντική, βαθμίδα του αθροιστή και κ_{16} είναι το κρατούμενο εξόδου.

 $\alpha + \beta$

$$\begin{array}{r}
 \alpha \quad 1111111110000110 \\
 \beta \quad + 1111111111000000 \\
 \hline
 \text{απ} \quad \textcircled{1}1111111101000110
 \end{array}$$


 το κρατούμενο εξόδου το αγνοούμε

$Y = \kappa_{16} \oplus \kappa_{15} = 1 \oplus 1 = 0$ επομένως δεν έχουμε υπερχείλιση.

Πράγματι $\alpha + \beta = -122 - 64 = -186_{(10)}$, ο οποίος είναι μεγαλύτερος του αριθμού -2^{15} που μπορεί να παρασταθεί με 16 δυαδικά ψηφία σε αναπαράσταση συμπληρώματος ως προς 2. Μπορούμε εύκολα να επιβεβαιώσουμε ότι το αποτέλεσμα της πρόσθεσης "1111111101000110" είναι ο αριθμός $-186_{(10)}$ σε αναπαράσταση συμπληρώματος ως προς 2.

 $\gamma + \delta$


$$\begin{array}{r}
 \gamma \quad 0000000001111100 \\
 \delta \quad + 0000000001000001 \\
 \hline
 \text{απ} \quad 0000000010111101
 \end{array}$$

$Y = \kappa_{16} \oplus \kappa_{15} = 0 \oplus 0 = 0$ επομένως δεν έχουμε υπερχείλιση.

Πράγματι $\gamma + \delta = 124 + 65 = 189_{(10)}$, ο οποίος είναι μικρότερος του αριθμού $2^{15} - 1$ που μπορεί να παρασταθεί με 16 δυαδικά ψηφία σε αναπαράσταση συμπληρώματος ως προς 2. Μπορούμε εύκολα να επιβεβαιώσουμε ότι το αποτέλεσμα της πρόσθεσης "0000000010111101" είναι ο αριθμός $189_{(10)}$ σε αναπαράσταση συμπληρώματος ως προς 2.

$\alpha + \gamma$

$$\begin{array}{r}
 \alpha \quad 1111111110000110 \\
 \gamma \quad + 0000000001111100 \\
 \hline
 \alpha\pi \quad \textcircled{1}0000000000000010
 \end{array}$$



 το κρατούμενο εξόδου το αγνοούμε

$Y = \kappa_{16} \oplus \kappa_{15} = 1 \oplus 1 = 0$ επομένως δεν έχουμε υπερχείλιση.

Πράγματι $\alpha + \gamma = -122_{(10)} + 124_{(10)} = 2_{(10)}$, το οποίο ισούται με το αποτέλεσμα που λάβαμε προσθέτοντας τους δυαδικούς αριθμούς σε αναπαράσταση συμπληρώματος ως προς 2.

$\beta + \delta$

$$\begin{array}{r}
 \alpha \quad 1111111111000000 \\
 \delta \quad + 0000000001000001 \\
 \hline
 \alpha\pi \quad \textcircled{1}0000000000000001
 \end{array}$$


 το κρατούμενο εξόδου το αγνοούμε

$Y = \kappa_{16} \oplus \kappa_{15} = 1 \oplus 1 = 0$ επομένως δεν έχουμε υπερχείλιση.

Πράγματι $\beta + \delta = -64_{(10)} + 65_{(10)} = 1_{(10)}$ το οποίο ισούται με το αποτέλεσμα που λάβαμε προσθέτοντας τους δυαδικούς αριθμούς σε αναπαράσταση συμπληρώματος ως προς 2.

Άσκηση 3.2

Ένας επεξεργαστής υποστηρίζει τόσο θετικά όσο και προσημασμένα ακέραια αριθμητικά δεδομένα μεγέθους μιας ψηφιολέξης (byte), μισής λέξης (δύο ψηφιολέξεις) και λέξης (τέσσερις ψηφιολέξεις). Όλες οι αριθμητικές πράξεις γίνονται με μία μονάδα των $v=32$ δυαδικών ψηφίων. Ο καταχωρητής κατάστασης μεταξύ των άλλων διαθέτει τις σημαίες K (κρατούμενο εξόδου), Y (υπερχείλιση, $Y = \kappa_v \oplus \kappa_{v-1}$) και M (μηδενικό αποτέλεσμα). Να δείξετε πως εκτελείται στην αριθμητική μονάδα του υπολογιστή κάθε μια από τις κάτωθι πράξεις, ποιες είναι οι τιμές που λαμβάνουν οι σημαίες K , Y και M σε κάθε περίπτωση και βασιζόμενοι στις τιμές των σημαιών να αποφανθείτε αν το αποτέλεσμα είναι σωστό.

Δεδομένα χωρίς πρόσημο στο δυαδικό:

μισής λέξης $\alpha = 1111111110000000$
 μιας λέξης $\beta = 11111111111111111000000000000000$
 $\gamma = 10000000000000000000000000000000$
 $\delta = 01110000000000000000000000000000$
 $\varepsilon = 00010000000000000000000000000000$

Δεδομένα σε μορφή συμπληρώματος ως προς 2:

μισής λέξης $\zeta = 1111111110000000$
 μιας λέξης $\eta = 11111111111111111100000000000000$
 $\theta = 10000000000000000000000000000000$

1^η πράξη: $\alpha + \beta$, 2^η πράξη: $\beta + \gamma$, 3^η πράξη: $\delta + \varepsilon$, 4^η πράξη $\zeta + \eta$ και 5^η πράξη: $\eta + \theta$.

Απάντηση

1^η πράξη: $\alpha + \beta$

Αφού οι αριθμοί είναι χωρίς πρόσημο ο α μετά την επέκταση περιοχής (προσήμου) θα γίνει

$\alpha = 000000000000000001111111110000000$

Οπότε

$$\begin{array}{r}
 \alpha \quad 0000000000000000001111111110000000 \\
 \beta \quad + 11111111111111111110000000000000 \\
 \hline
 \alpha\beta \quad \textcircled{1}0000000000000000001111111110000000
 \end{array}$$

$K=1, Y=0, M=0$

Αφού οι αριθμοί είναι χωρίς πρόσημο η τιμή του κρατούμενου δηλώνει την ύπαρξη υπερχειλίσης. Επειδή $K=1$ έχουμε υπερχειλίση, επομένως το αποτέλεσμα δεν χωράει σε 32 δυαδικά ψηφία.

2^η πράξη: β+γ

$$\begin{array}{r}
 \beta \quad 111111111111111111110000000000000000 \\
 \gamma \quad + 100000000000000000000000000000000000 \\
 \hline
 \alpha\pi \quad (1)011111111111111111110000000000000000
 \end{array}$$

K=1, Y= 1, M=0

Αφού οι αριθμοί είναι χωρίς πρόσημο η τιμή του κρατούμενου δηλώνει την ύπαρξη υπερχείλισης. Επειδή K=1 έχουμε υπερχείλιση, επομένως το αποτέλεσμα δεν χωράει σε 32 δυαδικά ψηφία.

3^η πράξη: δ+ε

$$\begin{array}{r}
 \delta \quad 011100000000000000000000000000000000 \\
 \varepsilon \quad + 000100000000000000000000000000000000 \\
 \hline
 \alpha\pi \quad 100000000000000000000000000000000000
 \end{array}$$

K=0, Y=1, M=0

Αφού οι αριθμοί είναι χωρίς πρόσημο η τιμή του κρατούμενου δηλώνει την ύπαρξη υπερχείλισης. Επειδή K=0 δεν έχουμε υπερχείλιση, επομένως το αποτέλεσμα είναι σωστό.

4^η πράξη: ζ+η

Αφού οι αριθμοί είναι σε αναπαράσταση συμπληρώματος ως προς 2 ο ζ μετά την επέκταση προσήμου θα γίνει ζ = 11111111111111111111111110000000

Οπότε

$$\begin{array}{r}
 \zeta \quad 111111111111111111111111111111110000000 \\
 \eta \quad + 11111111111111111111110000000000000000 \\
 \hline
 \alpha\pi \quad (1)1111111111111111111101111111110000000
 \end{array}$$

K=1, Y=0, M=0

γ. Ο καταχωρητής κατάστασης μεταξύ των άλλων διαθέτει τις σημαίες K (κρατούμενο εξόδου), Y (υπερχείλιση, $Y = \kappa_{v-1} \oplus \kappa_{v-2}$) και M (μηδενικό αποτέλεσμα). Υπάρχουν εντολές BRK (Branch if K), BRY (Branch if Y) και BRM (Branch if M). Ποια εντολή θα χρησιμοποιήσετε σε κάθε περίπτωση δεδομένων για την ανίχνευση υπερχείλισης;

δ. Πιστεύετε ότι το σύνολο των εντολών πρέπει να περιέχει διαφορετική εντολή για την αποθήκευση του περιεχομένου κάποιου καταχωρητή στη μνήμη ανάλογα του είδους του δεδομένου που περιέχει;

Να αιτιολογήσετε τις απαντήσεις σας.

Απάντηση

α. Για δεδομένα μιας ψηφιολέξης ή μισής λέξης χρειάζονται διαφορετικές εντολές για θετικά και προσημασμένα δεδομένα ώστε κατά την αποθήκευση στον καταχωρητή, ο οποίος είναι των 32 δυαδικών ψηφίων, να γίνει η σωστή επέκταση προσήμου. Για δεδομένα μιας λέξης, δηλαδή 4 ψηφιολέξεων, δεν χρειάζεται να γίνει επέκταση προσήμου, ότι διαβάζεται από τη μνήμη αποθηκεύεται στον καταχωρητή, επομένως δεν χρειάζονται διαφορετικές εντολές ανάγνωσης από τη μνήμη για θετικά και προσημασμένα δεδομένα μιας λέξης.

β. Δεν χρειάζονται διαφορετικές εντολές για δεδομένα μιας ψηφιολέξης, μισής λέξης και μιας λέξης διότι χρησιμοποιείται ή ίδια μονάδα για την εκτέλεση της αριθμητικής εντολής.

γ. Στην περίπτωση που τα δεδομένα που χρησιμοποιήσαμε είναι θετικοί αριθμοί θα χρησιμοποιήσουμε την εντολή BRK, ενώ αν τα δεδομένα είναι σε παράσταση συμπληρώματος ως προς 2 θα χρησιμοποιήσουμε την εντολή BRY.

δ. Χρειάζονται διαφορετικές εντολές για δεδομένα μιας ψηφιολέξης, μισής λέξης και μιας λέξης διότι στην πρώτη περίπτωση (αποθήκευση ψηφιολέξης) μόνο τα 8 λιγότερο σημαντικά δυαδικά ψηφία του καταχωρητή θα αποθηκευτούν σε μια θέση μνήμης, στην δεύτερη περίπτωση τα 16 λιγότερο σημαντικά δυαδικά ψηφία του καταχωρητή θα αποθηκευτούν σε δύο διαδοχικές θέσεις μνήμης και στην τρίτη περίπτωση όλο το περιεχόμενο του καταχωρητή θα αποθηκευτεί σε 4 διαδοχικές θέσεις μνήμης. Δεν χρειάζονται διαφορετικές εντολές για θετικά και προσημασμένα δεδομένα.

Άσκηση 3.4

Δίνονται οι αριθμοί κινητής υποδιαστολής σύμφωνα με το πρότυπο IEEE 754:

$$Y=00111111110000000000000000000000,$$

$$X=01001101100000100000000000000000$$

$$Z=11001101100000100000000000000000$$

Να υπολογίσετε την τιμή των εκφράσεων $Y+(X+Z)$ και $(Y+X)+Z$, να συγκρίνετε τα αποτελέσματα και να τα δικαιολογήσετε.

Απάντηση

Το πρόσημο Π_Y του Y είναι $\Pi_Y=0$, άρα ο αριθμός Y είναι θετικός.

Ο εκθέτης E_Y του Y είναι $E_Y=01111111$, επομένως $E_Y=127_{(10)}$. Ο συντελεστής Σ_Y του Y είναι $\Sigma_Y=1,110000000000000000000000$

Το πρόσημο Π_X του X είναι $\Pi_X=0$, άρα ο αριθμός X είναι θετικός.

Ο εκθέτης E_X του X είναι $E_X=10011011$, επομένως $E_X=155_{(10)}$. Ο συντελεστής Σ_X του X είναι $\Sigma_X=1,000001000000000000000000$

Το πρόσημο Π_Z του Z είναι $\Pi_Z=1$, άρα ο αριθμός Z είναι αρνητικός.

Ο εκθέτης E_Z του Z είναι $E_Z=10011011$, επομένως $E_Z=155_{(10)}$. Ο συντελεστής Σ_Z του Z είναι $\Sigma_Z=1,000001000000000000000000$

α. $Y+(X+Z)$

$$\begin{aligned} X+Z &= 2^{155} \times 1,000001000000000000000000 - 2^{155} \times 1,000001000000000000000000 = \\ &= 2^{155} \times 0,000000000000000000000000 \end{aligned}$$

Προφανώς αυτό το αποτέλεσμα ισούται με το μηδέν. Τα κυκλώματα που χρησιμοποιούνται για την εκτέλεση της πράξης, όταν ο εκθέτης είναι μικρότερος του 255 και ο συντελεστής είναι μηδέν, μηδενίζουν τον εκθέτη ώστε η μηδενική τιμή που θα αποθηκευτεί σ' ένα καταχωρητή να συμφωνεί με το πρότυπο 754. Επομένως,

$$X+Z = 2^0 \times 0,000000000000000000000000 = 0$$

$$\begin{aligned} Y+(X+Z) &= 2^{127} \times 1,110000000000000000000000 + \\ &\quad 2^0 \times 0,000000000000000000000000 = \\ &= 2^{127} \times 1,110000000000000000000000 + \\ &\quad 2^{127} \times 0,000000000000000000000000 = \\ &= 2^{127} \times 1,110000000000000000000000 = Y \end{aligned}$$

β. $(Y+X)+Z$

$$Y+X = 2^{127} \times 1,110000000000000000000000 + 2^{155} \times 1,000001000000000000000000$$

Για να προσθέσουμε δύο αριθμούς, αυτοί πρέπει να έχουν τον ίδιο εκθέτη. Επομένως ολισθαίνουμε τον αριθμό με τον μικρότερο εκθέτη προς τα δεξιά κατά $155-127=28$ θέσεις και αυξάνουμε τον εκθέτη του κατά 28. Τότε από τον

$2^{127} \times 1,110000000000000000000000$ λαμβάνουμε τον

$2^{155} \times 0,000000000000000000000000$ οπότε

$$\begin{aligned} Y+X &= 2^{155} \times 0,000000000000000000000000 + 2^{155} \times 1,000001000000000000000000 = \\ &= 2^{155} \times 1,000001000000000000000000 \end{aligned}$$

$$\begin{aligned} (Y+X)+Z &= 2^{155} \times 1,000001000000000000000000 - 2^{155} \times 1,000001000000000000000000 = \\ &= 2^{155} \times 0,000000000000000000000000 = 2^0 \times 0,000000000000000000000000. \end{aligned}$$

Παρατηρούμε ότι $Y+(X+Z) \neq (Y+X)+Z$. Ενώ στην άλγεβρα ισχύει η προσεταιριστική ιδιότητα στην πρόσθεση, παρατηρούμε ότι στις πράξεις μεταξύ αριθμών σε αναπαράσταση κινητής υποδιαστολής υπάρχουν περιπτώσεις που δεν ισχύει. Αυτό συμβαίνει όταν προσθέτουμε δύο μεγάλους αριθμούς αντίθετου προσήμου με ένα μικρό αριθμό.

Άσκηση 3.5 (3.2 βιβλίου)

Χρησιμοποιώντας πολυπλέκτες να σχεδιάσετε ολισθητή των τεσσάρων δυαδικών ψηφίων με δυνατότητα κυκλικής και αριθμητικής ολίσθησης προς τα δεξιά.

Απάντηση

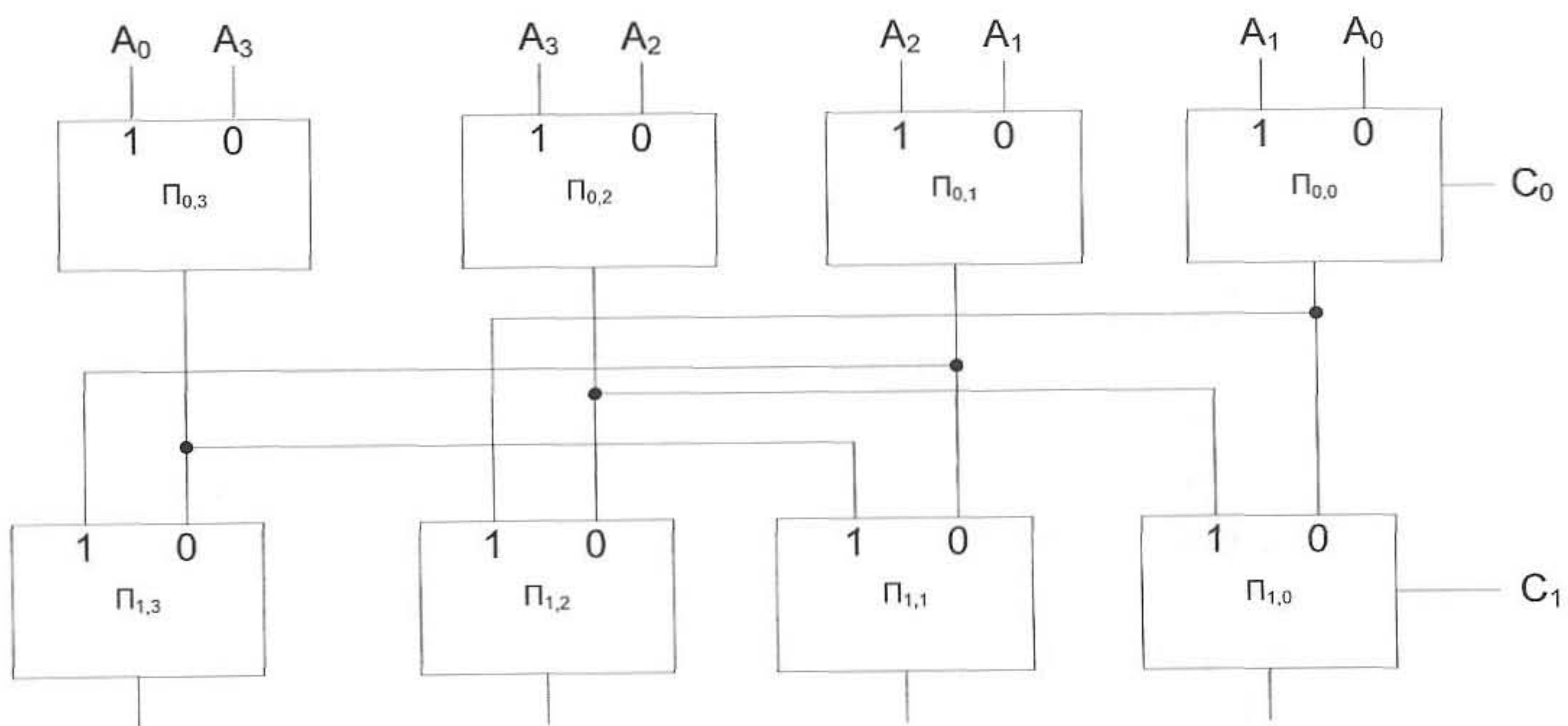
Αναφερθείτε στο βιβλίο «Αρχιτεκτονική Υπολογιστών, Δημήτρης Νικολός, Β. Γκιούρδας Εκδοτική», στη σελίδα 106 για συνδυαστικούς ολισθητές πολλών λειτουργιών.

Εφόσον επιθυμούμε να σχεδιάσουμε ένα ολισθητή με δύο λειτουργίες, κυκλική ολίσθηση και αριθμητική ολίσθηση, θα πρέπει να έχουμε μία είσοδο επιλογής, t . Εφόσον ο ολισθητής θα είναι των τεσσάρων δυαδικών ψηφίων, θα αποτελείται από δύο επίπεδα. Θα σχεδιάσουμε αρχικά ένα κύκλωμα κυκλικής ολίσθησης προς τα δεξιά και ένα κύκλωμα αριθμητικής ολίσθησης προς τα δεξιά. Στη συνέχεια θα δούμε πως παρατηρώντας τα δύο αυτά κυκλώματα μπορούμε να σχεδιάσουμε ένα κύκλωμα που ανάλογα της τιμής της εισόδου του t θα εκτελεί κυκλική ή αριθμητική ολίσθηση προς τα δεξιά.

Σχεδίαση κυκλώματος κυκλικής ολίσθησης.

Αφού το κύκλωμά μας θα υλοποιεί μόνο ένα είδος ολίσθησης κάθε επίπεδο θα αποτελείται μόνο από μια σειρά πολυπλεκτών. Οι πολυπλέκτες του πρώτου επιπέδου θα έχουν ως είσοδο επιλογής τη γραμμή C_0 , ενώ οι πολυπλέκτες του δεύτερου επιπέδου την C_1 . Όταν η γραμμή C_0 έχει την τιμή 0, τότε δεν πρέπει να γίνεται ολίσθηση σε αυτό το επίπεδο πολυπλεκτών, επομένως τις εισόδους των πολυπλεκτών που αντιστοιχούν στην τιμή $C_0 = 0$ τις οδηγούμε με την αρχική μας λέξη. Όταν το C_0 έχει την τιμή 1 τότε η λέξη εξόδου των πολυπλεκτών του πρώτου επιπέδου θα είναι η λέξη εισόδου ολισθημένη κατά μία θέση προς τα δεξιά. Το λιγότερο σημαντικό ψηφίο της λέξης εισόδου θα εμφανίζεται στην πιο αριστερή θέση της λέξης εξόδου των πολυπλεκτών. Επομένως τις εισόδους των πολυπλεκτών που αντιστοιχούν στην τιμή $C_0 = 1$ τις οδηγούμε με την αρχική μας λέξη ολισθημένη κατά μία θέση προς τα δεξιά και την πιο αριστερή είσοδο με το A_0 (βλέπε σχήμα 3.5.1)

Όσον αφορά το δεύτερο επίπεδο πολυπλεκτών, όταν η γραμμή C_1 έχει την τιμή 0, τότε δεν πρέπει να γίνεται ολίσθηση σε αυτό το επίπεδο πολυπλεκτών, επομένως τις εισόδους των πολυπλεκτών που αντιστοιχούν στην τιμή $C_1 = 0$ τις οδηγούμε από τις εξόδους των αντίστοιχων πολυπλεκτών του πρώτου επιπέδου. Όταν το C_1 έχει την τιμή 1 τότε η λέξη εξόδου των πολυπλεκτών του δεύτερου επιπέδου θα είναι η λέξη εξόδου των πολυπλεκτών του πρώτου επιπέδου ολισθημένη κατά δύο θέσεις προς τα δεξιά κυκλικά. Επομένως την είσοδο του πολυπλέκτη i που αντιστοιχεί στην τιμή $C_1 = 1$ την οδηγούμε από την έξοδο του πολυπλέκτη $i+2 \text{ modulo } 4$ (επειδή έχουμε 4 δυαδικά ψηφία) (βλέπε σχήμα 3.5.1).



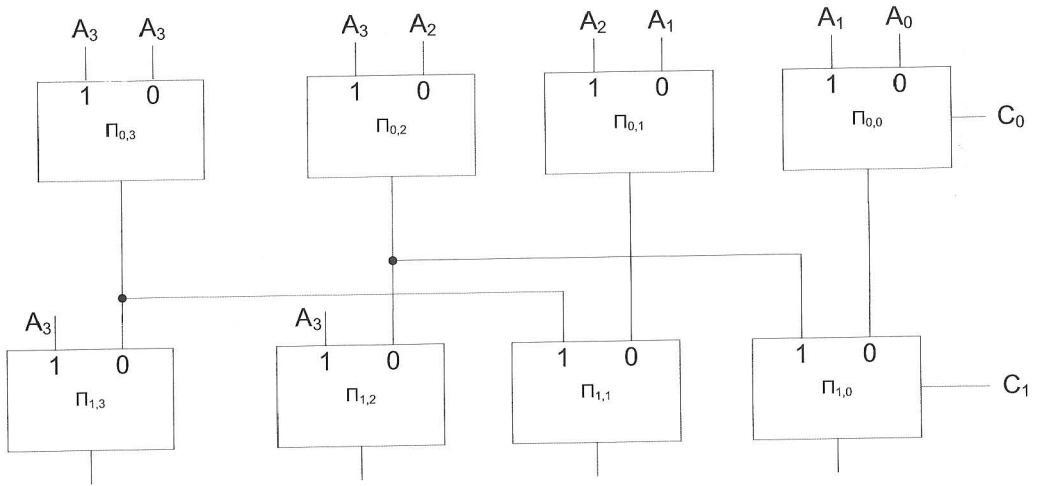
Σχήμα 3.5.1 Κύκλωμα κυκλικής ολίσθησης προς τα δεξιά.

Σχεδίαση κυκλώματος αριθμητικής ολίσθησης.

Αφού το κύκλωμά μας θα υλοποιεί μόνο ένα είδος ολίσθησης κάθε επίπεδο θα αποτελείται μόνο από μια σειρά πολυπλεκτών. Οι πολυπλέκτες του πρώτου επιπέδου θα έχουν ως είσοδο επιλογής τη γραμμή C_0 , ενώ οι πολυπλέκτες του δεύτερου επιπέδου την C_1 . Όταν η γραμμή C_0 έχει την τιμή 0, τότε δεν πρέπει να γίνεται ολίσθηση σε αυτό το επίπεδο πολυπλεκτών, επομένως τις εισόδους των πολυπλεκτών που αντιστοιχούν στην τιμή $C_0 = 0$ τις οδηγούμε με την αρχική μας λέξη. Όταν το C_0 έχει την τιμή 1 τότε η λέξη εξόδου των πολυπλεκτών του πρώτου επιπέδου θα είναι η λέξη εισόδου ολισθημένη κατά μία θέση προς τα δεξιά. Αφού έχουμε αριθμητική ολίσθηση προς τα δεξιά, στην πιο αριστερή θέση της λέξης εξόδου των πολυπλεκτών εμφανίζεται το πιο σημαντικό ψηφίο της αρχικής λέξης. Επομένως τις εισόδους των πολυπλεκτών που αντιστοιχούν στην τιμή $C_0 = 1$ τις οδηγούμε με την αρχική μας λέξη ολισθημένη κατά μία θέση προς τα δεξιά και την πιο αριστερή είσοδο με το A_3 (βλέπε σχήμα 3.5.2)

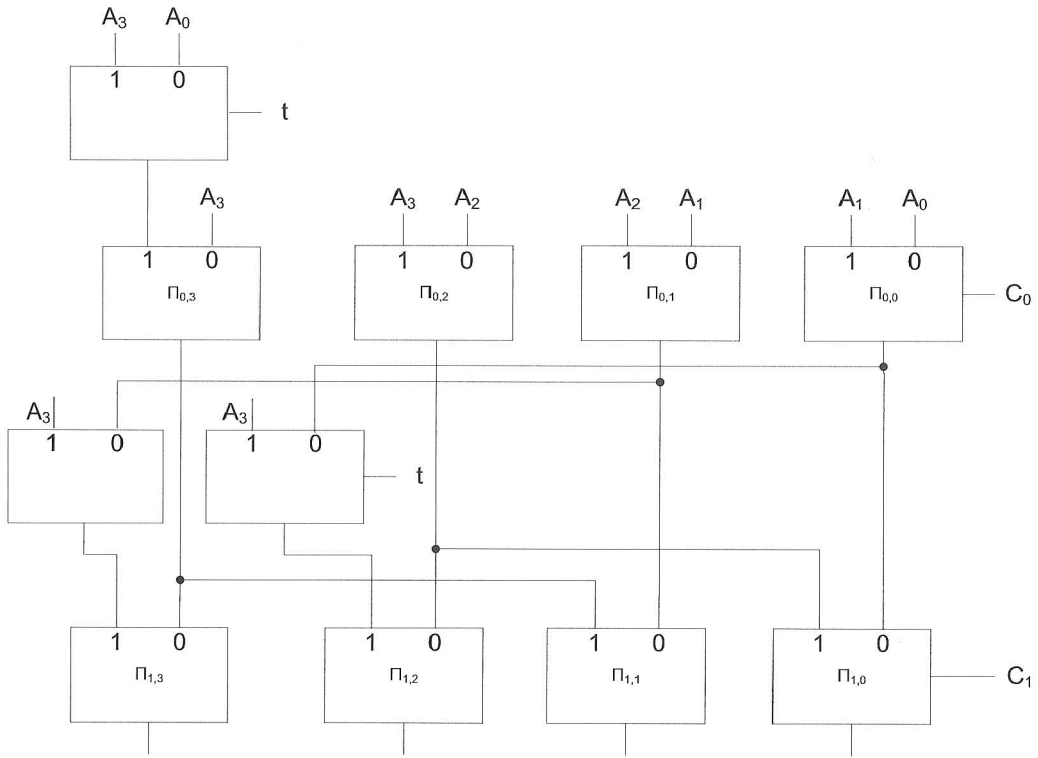
Όσον αφορά το δεύτερο επίπεδο πολυπλεκτών, όταν η γραμμή C_1 έχει την τιμή 0, τότε δεν πρέπει να γίνεται ολίσθηση σε αυτό το επίπεδο πολυπλεκτών, επομένως τις εισόδους των πολυπλεκτών που αντιστοιχούν στην τιμή $C_1 = 0$ τις οδηγούμε από τις εξόδους των αντίστοιχων πολυπλεκτών του πρώτου επιπέδου. Όταν το C_1 έχει την τιμή 1 τότε η λέξη εξόδου των πολυπλεκτών του δεύτερου επιπέδου θα είναι η λέξη εξόδου των πολυπλεκτών του πρώτου επιπέδου ολισθημένη κατά δύο θέσεις προς τα δεξιά. Στις εξόδους των δύο πολυπλεκτών που είναι στις δύο πιο αριστερές θέσεις θα πρέπει να εμφανίζεται το πιο σημαντικό ψηφίο της αρχικής λέξης, δηλαδή το A_3 (βλέπε σχήμα 3.5.2).

Παρατηρώντας το πρώτο επίπεδο πολυπλεκτών των σχημάτων 3.5.1 και 3.5.2 διαπιστώνουμε ότι στην περίπτωση του σχήματος 3.5.1 (κυκλική ολίσθηση προς τα δεξιά) η είσοδος του πολυπλέκτη $\Pi_{0,3}$ που αντιστοιχεί σε $C_0=1$ οδηγείται με το A_0 ενώ η αντίστοιχη είσοδος στο σχήμα 3.5.2 (αριθμητική ολίσθηση προς τα δεξιά) οδηγείται με το A_3 . Όλες οι άλλες εισόδους των πολυπλεκτών του πρώτου επιπέδου των σχημάτων 3.5.1 και 3.5.2 είναι ταυτοτικές. Επομένως ένα κύκλωμα για να μπορεί να χρησιμοποιηθεί είτε για κυκλική ολίσθηση προς τα δεξιά ($t=0$) είτε για αριθμητική ολίσθηση προς τα δεξιά ($t=1$), θα πρέπει την είσοδο που αντιστοιχεί σε $C_0=1$ να την οδηγήσουμε από την έξοδο ενός πολυπλέκτη 2-σε-1 με είσοδο ελέγχου t , που η είσοδός του που αντιστοιχεί σε $t=0$ να οδηγείται με το A_0 και η είσοδός του που αντιστοιχεί σε $t=1$ να οδηγείται με το A_3 .



Σχήμα 3.5.2 Κύκλωμα αριθμητικής ολίσθησης προς τα δεξιά.

Παρατηρώντας το δεύτερο επίπεδο πολυπλεκτών των σχημάτων 3.5.1 και 3.5.2 διαπιστώνουμε ότι στην περίπτωση του σχήματος 3.5.1 (κυκλική ολίσθηση προς τα δεξιά) οι είσοδοι των πολυπλεκτών $\Pi_{1,3}$ και $\Pi_{1,2}$ που αντιστοιχούν σε $C_0=1$ οδηγούνται από τις εξόδους των πολυπλεκτών $\Pi_{0,1}$ και $\Pi_{0,0}$ ενώ οι αντίστοιχες είσοδοι στο σχήμα 3.5.2 (αριθμητική ολίσθηση προς τα δεξιά) οδηγούνται με το A_3 . Επομένως οι είσοδοι των πολυπλεκτών $\Pi_{1,3}$ και $\Pi_{1,2}$ που αντιστοιχούν σε $C_1=1$ θα οδηγούνται από τις εξόδους δύο πολυπλεκτών 2-σε-1 με είσοδο ελέγχου t , που οι είσοδοί τους που αντιστοιχούν σε $t=0$ θα οδηγούνται από τις εξόδους των πολυπλεκτών $\Pi_{0,1}$ και $\Pi_{0,0}$, ενώ οι είσοδοί τους που αντιστοιχούν σε $t=1$ θα οδηγούνται με το A_3 . Όλες οι άλλες είσοδοι των πολυπλεκτών του δεύτερου επιπέδου των σχημάτων 3.5.1 και 3.5.2 είναι ταυτοτικές (σχήμα 3.5.3).



Σχήμα 3.5.3 Κύκλωμα κυκλικής ολίσθησης προς τα δεξιά για $t=0$ και αριθμητικής ολίσθησης προς τα δεξιά για $t=1$ κατά C_1C_0 θέσεις.

Άσκηση 3.6 (3.3 βιβλίου)

Να σχεδιάσετε αθροιστή πρόβλεψης κρατούμενων των 64 δυαδικών ψηφίων με την ελάχιστη καθυστέρηση σε επίπεδα πυλών, που καμία πύλη του να μην έχει περισσότερες από πέντε εισόδους.

Απάντηση

Έστω ότι $A = a_{v-1}a_{v-2}...a_1a_0$ και $B = b_{v-1}b_{v-2}...b_1b_0$ είναι δύο δυαδικοί αριθμοί των v δυαδικών ψηφίων και $S = s_{v-1}s_{v-2}...s_1s_0$ είναι το άθροισμά τους. Ορίζουμε τρεις όρους, τον όρο γένεσης κρατούμενου: $g_i = a_i \cdot b_i$, τον όρο διάδοσης κρατούμενου: $p_i = a_i + b_i$, και το ημιάθροισμα $h_i = a_i \oplus b_i$.

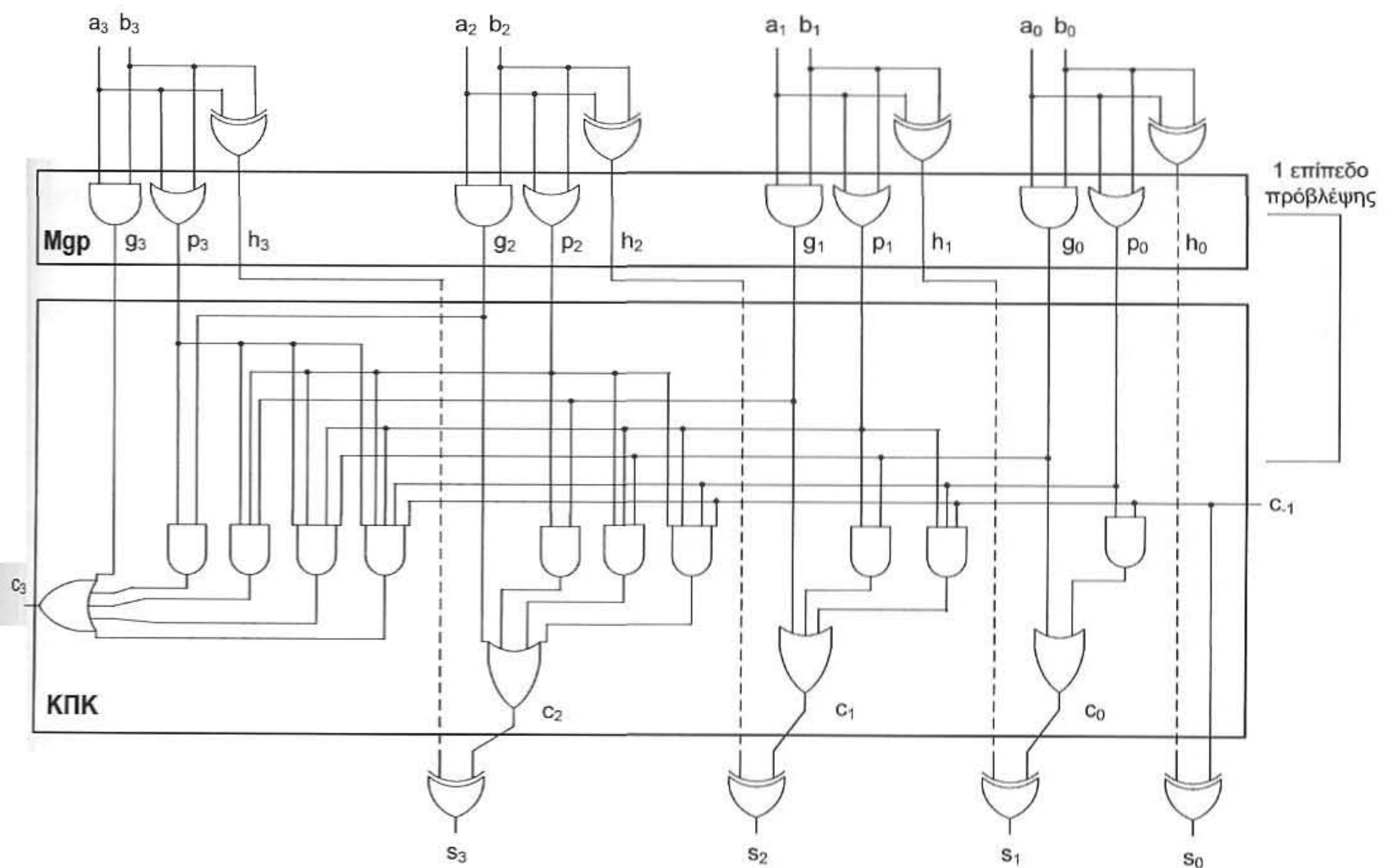
Είναι εύκολο να επιβεβαιώσουμε ότι για i από 0 έως και n έχουμε:

$$c_i = g_i + p_i \cdot c_{i-1}$$

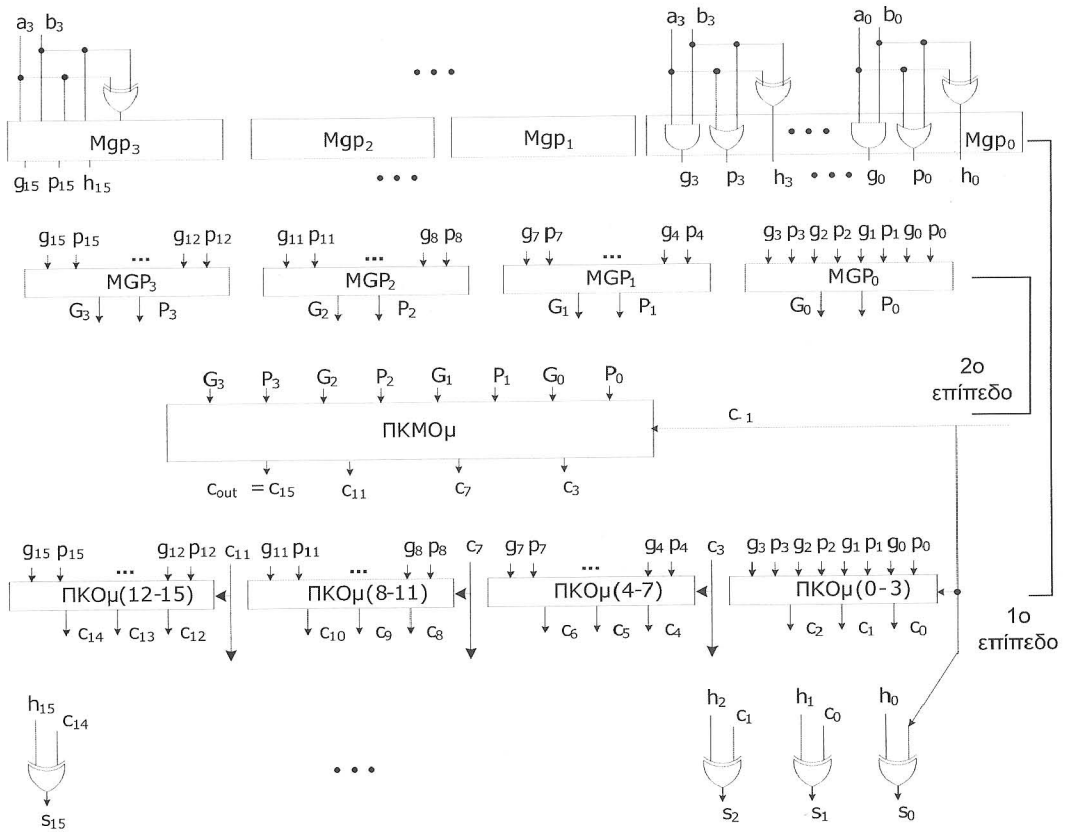
$$\text{και } s_i = h_i \oplus c_{i-1}$$

Πριν προχωρήσουμε στη σχεδίαση του ζητούμενου αθροιστή των 64 δυαδικών ψηφίων θα σχολιάσουμε τους αντίστοιχους αθροιστές πρόβλεψης κρατούμενου των 4 και 16 δυαδικών ψηφίων, η δομή των οποίων δίνεται στα σχήματα 3.6.1 και 3.6.2 (σχήματα 3.4 και 3.5 του βιβλίου Αρχιτεκτονική Υπολογιστών, Δημήτρης Νικολός, Εκδόσεις Β. Γκιούρδας Εκδοτική, 2008).

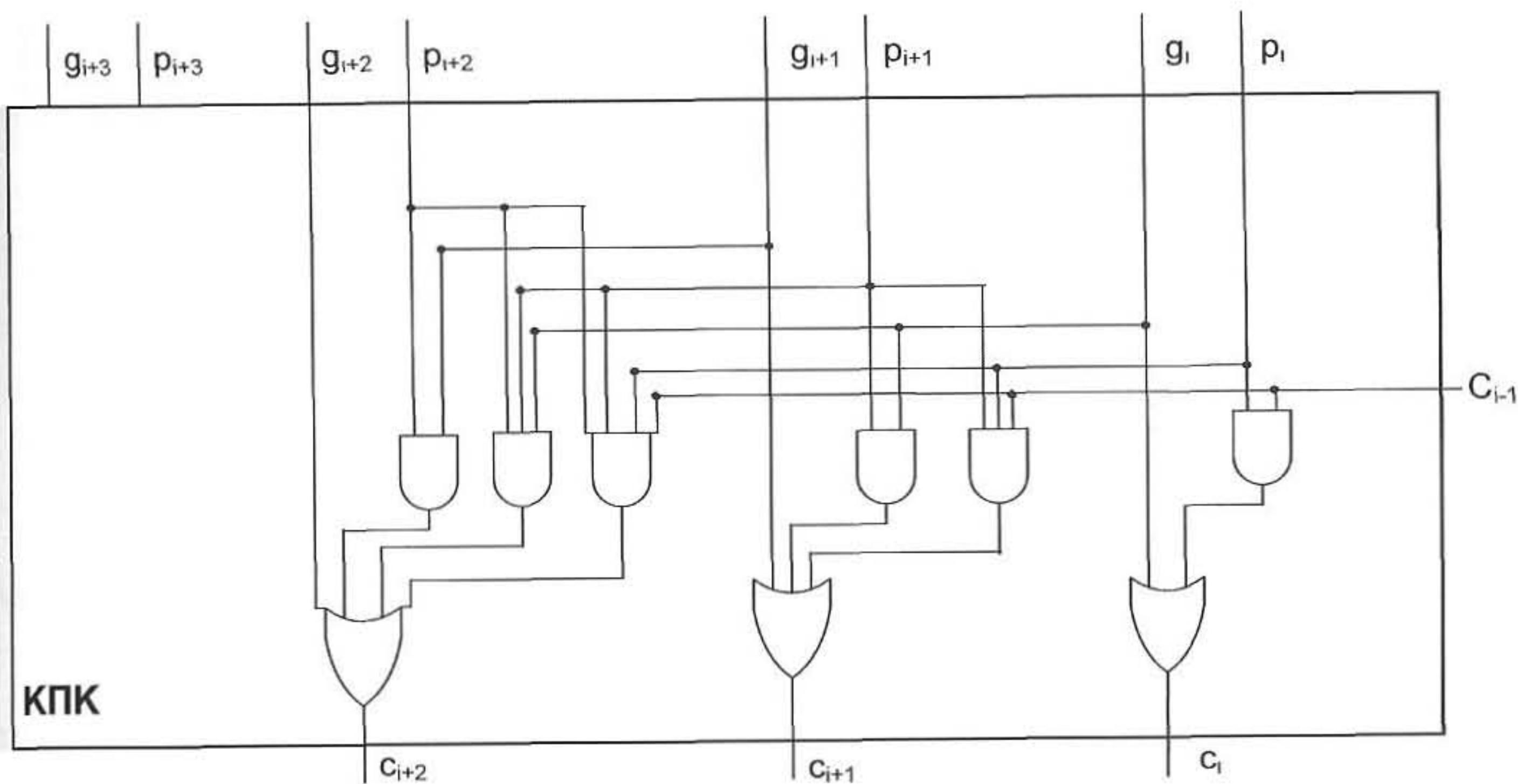
Από τα σχήματα 3.6.1 και 3.6.2 παρατηρούμε ότι στις εισόδους κάθε αθροιστή υπάρχει μια σειρά από n XOR πύλες για τον υπολογισμό των ημιαθροισμάτων και στις εξόδους μία σειρά από n XOR πύλες για τον υπολογισμό των δυαδικών ψηφίων του αθροίσματος. Ο αθροιστής του σχήματος 3.6.2 έχει 4 μονάδες Mgp_i , με $i=0, 1, 2$ και 3 , για τον υπολογισμό των p_i και g_i με $i=0, 1, \dots, 15$. Η μονάδα πρόβλεψης των κρατούμενων μέσα στην ομάδα $i-j$, ΠΚΟμ ($i-j$) έχει τη δομή που δίνεται στο σχήμα 3.6.3. Παρατηρούμε ότι έχει τη δομή της μονάδας ΚΠΚ του σχήματος 3.6.1 με μόνη διαφορά ότι δεν χρειάζεται το υποκύκλωμα υπολογισμού του πιο σημαντικού κρατούμενου της τετράδας διότι υπολογίζεται στη μονάδα ΠΚΜΟμ. Στον αθροιστή του σχήματος 3.6.2 έχουμε 4 μονάδες ΠΚΟμ($i-j$), συγκεκριμένα ΠΚΟμ(0-3), ΠΚΟμ(4-7) ΠΚΟμ(8-11) και ΠΚΟμ(12-15). Γενικότερα μπορούμε να παρατηρήσουμε ότι κάθε μονάδα του αθροιστή του σχήματος 3.6.1 εμφανίζεται τετραπλάσιο αριθμό φορές στον αθροιστή του σχήματος 3.6.2. Επίσης στον αθροιστή του σχήματος 3.6.2 χρησιμοποιούνται οι μονάδες MGP_i με $i=0, 1, 2$ και 3 που υπολογίζουν τα σήματα διάδοσης κρατούμενου P_i και γένεσης κρατούμενου G_i μέσα στην ομάδα i καθώς και η μονάδα ΠΚΜΟμ πρόβλεψης κρατούμενων μεταξύ των ομάδων. Η μονάδα ΠΚΜΟμ έχει την δομή της μονάδας ΚΠΚ του αθροιστή του σχήματος 3.6.1.



Σχήμα 3.6.1 Αθροιστής πρόβλεψης κρατούμενου των 4 δυαδικών ψηφίων. Mgp είναι η μονάδα υπολογισμού των σημάτων διάδοσης κρατούμενου p_i και γένεσης κρατούμενου g_i . ΚΠΚ είναι το κύκλωμα πρόβλεψης κρατούμενων.



Σχήμα 3.6.2. Αθροιστής δύο επιπέδων πρόβλεψης κρατούμενου των 16 δυαδικών ψηφίων. Mgp είναι η μονάδα υπολογισμού των σημάτων διάδοσης κρατούμενου p_i και γένεσης κρατούμενου g_i . MGP_i είναι η μονάδα υπολογισμού των G_i και P_i της ομάδας i , δηλαδή υλοποιεί τις σχέσεις (3,6) του βιβλίου. $ΠΚΟΜ$ είναι η μονάδα πρόβλεψης κρατούμενων μεταξύ ομάδων, δηλαδή υλοποιεί τις σχέσεις (3.7) του βιβλίου. $ΠΚΟΜ(i-j)$ είναι η μονάδα πρόβλεψης των κρατούμενων μέσα στην ομάδα $i-j$, δηλαδή υλοποιεί τις σχέσεις (3.8) του βιβλίου. Επειδή τα κρατούμενα c_3, c_7, c_{11} και c_{15} παράγονται από την υπομονάδα $ΠΚΟΜ$ δε χρειάζεται να επαναυπολογιστούν από τις υπομονάδες $ΠΚΟΜ(0-3), ΠΚΟΜ(4-7), ΠΚΟΜ(8-11)$ και $ΠΚΟΜ(12-15)$ αντίστοιχα.



Σχήμα 3.6.3 Η δομή της μονάδας πρόβλεψης των κρατούμενων μέσα στην ομάδα $i-j$, ΠΚΟμ $(i-j)$, με $j=i+3$.

Από το σχήμα 3.6.1 παρατηρούμε ότι για να υπολογιστεί το κρατούμενο c_i χρησιμοποιούνται πύλες των $i+1$ εισόδων. Άρα σε ένα αθροιστή των n εισόδων θα απαιτούνται πύλες των $n+1$ εισόδων. Επομένως αφού θέλουμε καμία πύλη να μην έχει περισσότερες από πέντε εισόδους, πρέπει να σχεδιάσουμε αθροιστές με περισσότερα επίπεδα πρόβλεψης κρατούμενου. Σε κάθε επίπεδο υπολογισμού κρατούμενων οι είσοδοι μοιράζονται σε ομάδες το πολύ των τεσσάρων στοιχείων. Για το σκοπό αυτό θα χρησιμοποιήσουμε τους όρους G_i και P_i . Όταν $P_i = 1$, τότε εάν το κρατούμενο εισόδου στην ομάδα i έχει την τιμή ένα, η τιμή αυτή θα διαδοθεί στο κρατούμενο εξόδου της ομάδας. Το G_i λαμβάνει την τιμή ένα όταν στην ομάδα i γεννιέται ένα κρατούμενο και διαδίδεται μέχρι το κρατούμενο εξόδου της ομάδας. Για να σχεδιάσουμε ένα αθροιστή πρόβλεψης κρατούμενου των 64 δυαδικών ψηφίων, έτσι ώστε κάθε ομάδα να έχει μέχρι τέσσερα στοιχεία θα έχουμε $\log_4 64 = 3$ επίπεδα πρόβλεψης κρατούμενου. Θα θεωρήσουμε ομάδες των τεσσάρων δυαδικών ψηφίων.

$$P_0 = p_3 \cdot p_2 \cdot p_1 \cdot p_0$$

$$G_0 = g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0$$

$$P_1 = p_7 \cdot p_6 \cdot p_5 \cdot p_4$$

$$G_1 = g_7 + p_7 \cdot g_6 + p_7 \cdot p_6 \cdot g_5 + p_7 \cdot p_6 \cdot p_5 \cdot g_4$$

$$P_2 = p_{11} \cdot p_{10} \cdot p_9 \cdot p_8$$

$$G_2 = g_{11} + p_{11} \cdot g_{10} + p_{11} \cdot p_{10} \cdot g_9 + p_{11} \cdot p_{10} \cdot p_9 \cdot g_8$$

$$P_3 = p_{15} \cdot p_{14} \cdot p_{13} \cdot p_{12}$$

$$G_3 = g_{15} + p_{15} \cdot g_{14} + p_{15} \cdot p_{14} \cdot g_{13} + p_{15} \cdot p_{14} \cdot p_{13} \cdot g_{12}$$

$$P_4 = p_{19} \cdot p_{18} \cdot p_{17} \cdot p_{16}$$

$$G_4 = g_{19} + p_{19} \cdot g_{18} + p_{19} \cdot p_{18} \cdot g_{17} + p_{19} \cdot p_{18} \cdot p_{17} \cdot g_{16}$$

$$P_5 = p_{23} \cdot p_{22} \cdot p_{21} \cdot p_{20}$$

$$G_5 = g_{23} + p_{23} \cdot g_{22} + p_{23} \cdot p_{22} \cdot g_{21} + p_{23} \cdot p_{22} \cdot p_{21} \cdot g_{20}$$

$$P_6 = p_{27} \cdot p_{26} \cdot p_{25} \cdot p_{24}$$

$$G_6 = g_{27} + p_{27} \cdot g_{26} + p_{27} \cdot p_{26} \cdot g_{25} + p_{27} \cdot p_{26} \cdot p_{25} \cdot g_{24}$$

$$P_7 = p_{31} \cdot p_{30} \cdot p_{29} \cdot p_{28}$$

$$G_7 = g_{31} + p_{31} \cdot g_{30} + p_{31} \cdot p_{30} \cdot g_{29} + p_{31} \cdot p_{30} \cdot p_{29} \cdot g_{28}$$

$$P_8 = p_{35} \cdot p_{34} \cdot p_{33} \cdot p_{32}$$

$$G_8 = g_{35} + p_{35} \cdot g_{34} + p_{35} \cdot p_{34} \cdot g_{33} + p_{35} \cdot p_{34} \cdot p_{33} \cdot g_{32}$$

$$P_9 = p_{39} \cdot p_{38} \cdot p_{37} \cdot p_{36}$$

$$G_9 = g_{39} + p_{39} \cdot g_{38} + p_{39} \cdot p_{38} \cdot g_{37} + p_{39} \cdot p_{38} \cdot p_{37} \cdot g_{36}$$

$$P_{10} = p_{43} \cdot p_{42} \cdot p_{41} \cdot p_{40}$$

$$G_{10} = g_{43} + p_{43} \cdot g_{42} + p_{43} \cdot p_{42} \cdot g_{41} + p_{43} \cdot p_{42} \cdot p_{41} \cdot g_{40}$$

$$P_{11} = p_{47} \cdot p_{46} \cdot p_{45} \cdot p_{44}$$

$$G_{11} = g_{47} + p_{47} \cdot g_{46} + p_{47} \cdot p_{46} \cdot g_{45} + p_{47} \cdot p_{46} \cdot p_{45} \cdot g_{44}$$

$$P_{12} = p_{51} \cdot p_{50} \cdot p_{49} \cdot p_{48}$$

$$G_{12} = g_{51} + p_{51} \cdot g_{50} + p_{51} \cdot p_{50} \cdot g_{49} + p_{51} \cdot p_{50} \cdot p_{49} \cdot g_{48}$$

$$P_{13} = p_{55} \cdot p_{54} \cdot p_{53} \cdot p_{52}$$

$$G_{13} = g_{55} + p_{55} \cdot g_{54} + p_{55} \cdot p_{54} \cdot g_{53} + p_{55} \cdot p_{54} \cdot p_{53} \cdot g_{52}$$

$$P_{14} = p_{59} \cdot p_{58} \cdot p_{57} \cdot p_{56}$$

$$G_{14} = g_{59} + p_{59} \cdot g_{58} + p_{59} \cdot p_{58} \cdot g_{57} + p_{59} \cdot p_{58} \cdot p_{57} \cdot g_{56}$$

$$P_{15} = p_{63} \cdot p_{62} \cdot p_{61} \cdot p_{60}$$

$$G_{15} = g_{63} + p_{63} \cdot g_{62} + p_{63} \cdot p_{62} \cdot g_{61} + p_{63} \cdot p_{62} \cdot p_{61} \cdot g_{60}$$

Ο υπολογισμός των G_i και P_i , με $i=0, 1, 2, 3, \dots, 15$, γίνεται με δεκαέξι μονάδες MGP_i .

Η πρόβλεψη κρατούμενων μεταξύ των ομάδων θα μπορούσε να γίνει βάσει των επόμενων σχέσεων:

$$gc_0 = G_0 + P_0 \cdot c_{-1}$$

$$gc_1 = G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot c_{-1}$$

$$gc_2 = G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot c_{-1}$$

$$gc_3 = G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot c_{-1}$$

.

.

.

$$\begin{aligned} gc_{15} = & G_{15} + P_{15} \cdot G_{14} + P_{15} \cdot P_{14} \cdot G_{13} + P_{15} \cdot P_{14} \cdot P_{13} \cdot G_{12} + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot G_{11} + \\ & + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot G_{10} + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot G_9 + \\ & + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot G_8 + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot G_7 + \\ & + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot G_6 + \\ & + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot G_5 + \\ & + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot G_4 + \\ & + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot G_3 + \\ & + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot G_2 + \\ & + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot G_1 + \\ & + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot G_0 + \\ & + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot c_{-1} \end{aligned}$$

Παρατηρούμε όμως ότι σ' αυτή την περίπτωση θα έπρεπε να χρησιμοποιήσουμε πύλες με πολύ περισσότερες από πέντε εισόδους. Επομένως θα πρέπει να χρησιμοποιήσουμε μια ομαδοποίηση των G_i και P_i ως ακολούθως.

$$GP_0 = P_3 \cdot P_2 \cdot P_1 \cdot P_0$$

$$GG_0 = G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0$$

$$GP_1 = P_7 \cdot P_6 \cdot P_5 \cdot P_4$$

$$GG_1 = G_7 + P_7 \cdot G_6 + P_7 \cdot P_6 \cdot G_5 + P_7 \cdot P_6 \cdot P_5 \cdot G_4$$

$$GP_2 = P_{11} \cdot P_{10} \cdot P_9 \cdot P_8$$

$$GG_2 = G_{11} + P_{11} \cdot G_{10} + P_{11} \cdot P_{10} \cdot G_9 + P_{11} \cdot P_{10} \cdot P_9 \cdot G_8$$

$$GP_3 = P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12}$$

$$GG_3 = G_{15} + P_{15} \cdot G_{14} + P_{15} \cdot P_{14} \cdot G_{13} + P_{15} \cdot P_{14} \cdot P_{13} \cdot G_{12}$$

Ο υπολογισμός των GG_i και GP_i γίνεται με τις μονάδες MGGP_i με $i=0, 1, 2$ και 3 .

Παρατηρούμε ότι

$$GP_0 = P_3 \cdot P_2 \cdot P_1 \cdot P_0 = p_{15} \cdot p_{14} \cdot p_{13} \cdot p_{12} \cdot p_{11} \cdot p_{10} \cdot p_9 \cdot p_8 \cdot p_7 \cdot p_6 \cdot p_5 \cdot p_4 \cdot p_3 \cdot p_2 \cdot p_1 \cdot p_0$$

και

$$\begin{aligned} GG_0 &= G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 = \\ &= g_{15} + p_{15} \cdot g_{14} + p_{15} \cdot p_{14} \cdot g_{13} + p_{15} \cdot p_{14} \cdot p_{13} \cdot g_{12} + \\ &+ P_3 \cdot (g_{11} + p_{11} \cdot g_{10} + p_{11} \cdot p_{10} \cdot g_9 + p_{11} \cdot p_{10} \cdot p_9 \cdot g_8) + \\ &+ P_3 \cdot P_2 \cdot (g_7 + p_7 \cdot g_6 + p_7 \cdot p_6 \cdot g_5 + p_7 \cdot p_6 \cdot p_5 \cdot g_4) + \\ &+ P_3 \cdot P_2 \cdot P_1 \cdot (g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0) = \\ &= g_{15} + p_{15} \cdot g_{14} + p_{15} \cdot p_{14} \cdot g_{13} + p_{15} \cdot p_{14} \cdot p_{13} \cdot g_{12} + \\ &+ p_{15} \cdot p_{14} \cdot p_{13} \cdot p_{12} \cdot (g_{11} + p_{11} \cdot g_{10} + p_{11} \cdot p_{10} \cdot g_9 + p_{11} \cdot p_{10} \cdot p_9 \cdot g_8) + \\ &+ p_{15} \cdot p_{14} \cdot p_{13} \cdot p_{12} \cdot p_{11} \cdot p_{10} \cdot p_9 \cdot p_8 \cdot (g_7 + p_7 \cdot g_6 + p_7 \cdot p_6 \cdot g_5 + p_7 \cdot p_6 \cdot p_5 \cdot g_4) + \\ &+ p_{15} \cdot p_{14} \cdot p_{13} \cdot p_{12} \cdot p_{11} \cdot p_{10} \cdot p_9 \cdot p_8 \cdot p_7 \cdot p_6 \cdot p_5 \cdot p_4 \cdot (g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + \\ &\quad + p_3 \cdot p_2 \cdot p_1 \cdot g_0) \end{aligned}$$

Από τις παραπάνω σχέσεις συνεπάγεται ότι όταν $GP_i = 1$, τότε εάν το κρατούμενο εισόδου στην ομάδα i έχει την τιμή ένα, η τιμή αυτή θα διαδοθεί στο κρατούμενο εξόδου της ομάδας. Το GG_i λαμβάνει την τιμή ένα όταν στην ομάδα i γεννιέται ένα κρατούμενο και διαδίδεται μέχρι το κρατούμενο εξόδου της ομάδας.

Από τα GG_i και GP_i , με $i=0, 1, 2$ και 3 , μπορούμε να προβλέψουμε τα κρατούμενα μεταξύ των τεσσάρων ομάδων, που είναι των δεκαέξι δυαδικών ψηφίων η κάθε μία, χρησιμοποιώντας τις ακόλουθες σχέσεις.

$$Gc_0 = GG_0 + GP_0 \cdot c_{-1}$$

$$Gc_1 = GG_1 + GP_1 \cdot GG_0 + GP_1 \cdot GP_0 \cdot c_{-1}$$

$$Gc_2 = GG_2 + GP_2 \cdot GG_1 + GP_2 \cdot GP_1 \cdot GG_0 + GP_2 \cdot GP_1 \cdot GP_0 \cdot c_{-1}$$

$$Gc_3 = GG_3 + GP_3 \cdot GG_2 + GP_3 \cdot GP_2 \cdot GG_1 + GP_3 \cdot GP_2 \cdot GP_1 \cdot GG_0 + \\ + GP_3 \cdot GP_2 \cdot GP_1 \cdot GP_0 \cdot c_{-1}$$

Έχουμε υπολογίσει ήδη τα GG_0 και GP_0 ως συνάρτηση των g_i και p_i , οπότε αντικαθιστώντας παίρνουμε

$$Gc_0 = \\ = g_{15} + p_{15} \cdot g_{14} + p_{15} \cdot p_{14} \cdot g_{13} + p_{15} \cdot p_{14} \cdot p_{13} \cdot g_{12} + \\ + p_{15} \cdot p_{14} \cdot p_{13} \cdot p_{12} \cdot (g_{11} + p_{11} \cdot g_{10} + p_{11} \cdot p_{10} \cdot g_9 + p_{11} \cdot p_{10} \cdot p_9 \cdot g_8) + \\ + p_{15} \cdot p_{14} \cdot p_{13} \cdot p_{12} \cdot p_{11} \cdot p_{10} \cdot p_9 \cdot p_8 \cdot (g_7 + p_7 \cdot g_6 + p_7 \cdot p_6 \cdot g_5 + p_7 \cdot p_6 \cdot p_5 \cdot g_4) + \\ + p_{15} \cdot p_{14} \cdot p_{13} \cdot p_{12} \cdot p_{11} \cdot p_{10} \cdot p_9 \cdot p_8 \cdot p_7 \cdot p_6 \cdot p_5 \cdot p_4 \cdot (g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0) + \\ + p_{15} \cdot p_{14} \cdot p_{13} \cdot p_{12} \cdot p_{11} \cdot p_{10} \cdot p_9 \cdot p_8 \cdot p_7 \cdot p_6 \cdot p_5 \cdot p_4 \cdot p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot c_{-1}$$

το οποίο είναι το c_{15} . Με τον ίδιο τρόπο μπορούμε να δούμε ότι $Gc_1 = c_{31}$, $Gc_2 = c_{47}$ και $Gc_3 = c_{63}$. Ο υπολογισμός των c_{15} , c_{31} , c_{47} και c_{63} γίνεται με την μονάδα ΠΚΜΟμ GG&GP.

Η πρόβλεψη κρατούμενων μεταξύ των ομάδων των τεσσάρων δυαδικών ψηφίων γίνεται βάσει των επόμενων σχέσεων με τη μονάδα ΠΚΜΟμ G&P:

$$gc_0 = G_0 + P_0 \cdot c_{-1}$$

$$gc_1 = G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot c_{-1}$$

$$gc_2 = G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot c_{-1}$$

$$gc_3 = G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot c_{-1}$$

$$gc_4 = G_4 + P_4 \cdot G_3 + P_4 \cdot P_3 \cdot G_2 + P_4 \cdot P_3 \cdot P_2 \cdot G_1 + P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot c_{-1}$$

$$gc_5 = G_5 + P_5 \cdot G_4 + P_5 \cdot P_4 \cdot G_3 + P_5 \cdot P_4 \cdot P_3 \cdot G_2 + P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot G_1 + \\ + P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot c_{-1}$$

$$gc_6 = G_6 + P_6 \cdot G_5 + P_6 \cdot P_5 \cdot G_4 + P_6 \cdot P_5 \cdot P_4 \cdot G_3 + P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot G_2 + \\ + P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot G_1 + P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot c_{-1}$$

$$gc_7 = G_7 + P_7 \cdot G_6 + P_7 \cdot P_6 \cdot G_5 + P_7 \cdot P_6 \cdot P_5 \cdot G_4 + P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot G_3 + P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot G_2 + \\ + P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot G_1 + P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot G_0 + \\ + P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot c_{-1}$$

.

.

.

$$gc_{15} = G_{15} + P_{15} \cdot G_{14} + P_{15} \cdot P_{14} \cdot G_{13} + P_{15} \cdot P_{14} \cdot P_{13} \cdot G_{12} + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot G_{11} + \\ + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot G_{10} + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot G_9 + \\ + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot G_8 + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot G_7 + \\ + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot G_6 + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot G_5 + \\ + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot G_4 + \\ + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot G_3 + \\ + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot G_2 + \\ + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot G_1 + \\ + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot G_0 + \\ + P_{15} \cdot P_{14} \cdot P_{13} \cdot P_{12} \cdot P_{11} \cdot P_{10} \cdot P_9 \cdot P_8 \cdot P_7 \cdot P_6 \cdot P_5 \cdot P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot c_{-1}$$

Αν αντικαταστήσουμε τα G_0 και P_0 θα πάρουμε

$$gc_0 = (g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0) + (p_3 \cdot p_2 \cdot p_1 \cdot p_0) \cdot c_{-1} \text{ το οποίο είναι } \\ \text{το } c_3. \text{ Με τον ίδιο τρόπο μπορούμε να δούμε ότι } gc_1 = c_7, gc_2 = c_{11}, gc_3 = c_{15}, gc_4 = c_{19}, \\ gc_5 = c_{23}, gc_6 = c_{27}, gc_7 = c_{31}, gc_8 = c_{35}, gc_9 = c_{39}, gc_{10} = c_{43}, gc_{11} = c_{47}, gc_{12} = c_{51}, \\ gc_{13} = c_{55}, gc_{14} = c_{59} \text{ και } gc_{15} = c_{63}.$$

Τα κρατούμενα c_{15} , c_{31} , c_{47} και c_{63} δε χρειάζεται να επαναυπολογιστούν από τις μονάδες ΠΚΜΟμG&P διότι έχουν υπολογιστεί από τη μονάδα ΠΚΜΟμ GG&GP.

Ο υπολογισμός των κρατούμενων c_i, c_{i+1}, c_{i+2} και c_{i+3} για $i = 0, 4, 8, 12, \dots, 56, 60$ γίνεται από τις επόμενες σχέσεις με τις μονάδες ΠΚΟμ (i-j).

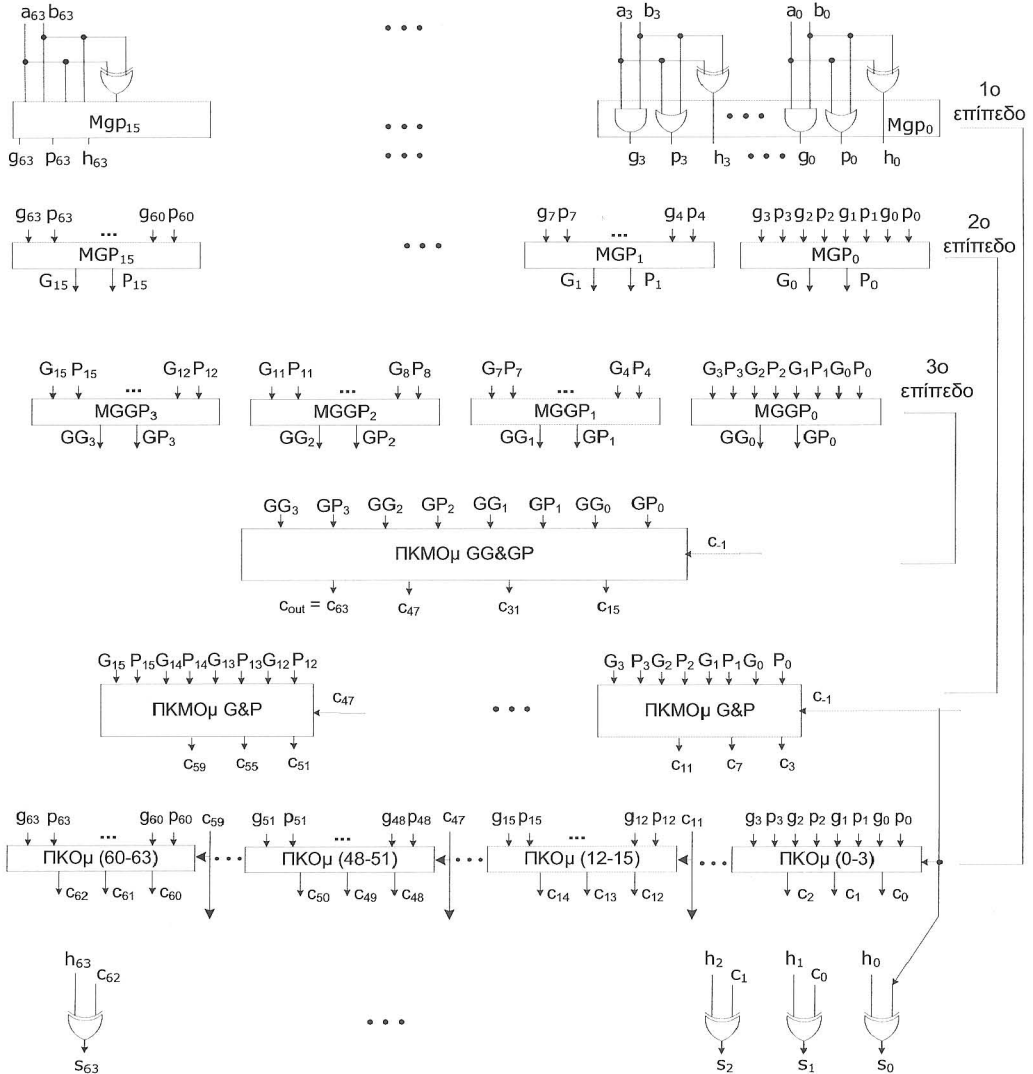
$$c_i = g_i + p_i \cdot c_{i-1}$$

$$c_{i+1} = g_{i+1} + p_{i+1} \cdot g_i + p_{i+1} \cdot p_i \cdot c_{i-1}$$

$$c_{i+2} = g_{i+2} + p_{i+2} \cdot g_{i+1} + p_{i+2} \cdot p_{i+1} \cdot g_i + p_{i+2} \cdot p_{i+1} \cdot p_i \cdot c_{i-1} \text{ και}$$

$$c_{i+3} = g_{i+3} + p_{i+3} \cdot g_{i+2} + p_{i+3} \cdot p_{i+2} \cdot g_{i+1} + p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot g_i + p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot p_i \cdot c_{i-1}$$

Ο αθροιστής τριών επιπέδων πρόβλεψης κρατούμενου των 64 δυαδικών ψηφίων δίνεται στο σχήμα 3.6.4.



Σχήμα 3.6.4 Αθροιστής τριών επιπέδων πρόβλεψης κρατούμενων των 64 δυαδικών ψηφίων.

Άσκηση 3.7 (3.4 βιβλίου)

Να σχεδιάσετε πολλαπλασιαστή διατήρησης κρατούμενου για μη προσημασμένους αριθμούς των 8 δυαδικών ψηφίων, του οποίου η τελευταία βαθμίδα άθροισης υλοποιείται από δύο αθροιστές πρόβλεψης κρατούμενου των τεσσάρων δυαδικών ψηφίων ο κάθε ένας και διάδοσης κρατούμενου μεταξύ αυτών. Να υπολογίσετε τη μέγιστη καθυστέρησή του και να τη συγκρίνετε με την καθυστέρηση του αντίστοιχου πολλαπλασιαστή διάδοσης κρατούμενου.

Απάντηση

Όταν έχουμε να προσθέσουμε n αριθμούς, τότε αντί να προσθέτουμε τους δύο από αυτούς και στη συνέχεια το αποτέλεσμα με τον τρίτο, έως ότου να φτάσουμε στο ολικό άθροισμα, έχουμε μικρότερη καθυστέρηση αν χρησιμοποιήσουμε αθροιστές διατήρησης του κρατούμενου (carry-save adders). Σ' αυτούς, σε κάθε επίπεδο άθροισης, εκτός από το τελευταίο, το κρατούμενο κάθε βαθμίδας δεν οδηγείται στον επόμενο ΠΑ ή ΗΑ αλλά σε έναν ΠΑ ή ΗΑ της επόμενης βαθμίδας άθροισης που έχει το ίδιο βάρος. Στο τελευταίο επίπεδο είμαστε υποχρεωμένοι να αθροίσουμε δύο ποσότητες χωρίς να έχουμε τη δυνατότητα διάδοσης των κρατούμενων προς επόμενο επίπεδο, μιας και τέτοιο δεν υπάρχει. Οι πολλαπλασιαστές αριθμών χωρίς πρόσημο που προκύπτουν όταν για την πρόσθεση των μερικών γινομένων χρησιμοποιείται αυτή η τεχνική, καλούνται πολλαπλασιαστές διατήρησης κρατούμενου (carry-save multipliers) («Αρχιτεκτονική Υπολογιστών, Δημήτρης Νικολός, Β. Γκιούρδας Εκδοτική», Σελίδα 115).

Στο σχήμα 3.7.1 δίνεται πολλαπλασιαστής διατήρησης κρατούμενου για μη προσημασμένους αριθμούς των 8 δυαδικών ψηφίων, του οποίου η τελευταία βαθμίδα άθροισης υλοποιείται από δύο αθροιστές πρόβλεψης κρατούμενου των τεσσάρων δυαδικών ψηφίων ο κάθε ένας και διάδοσης κρατούμενου μεταξύ αυτών.

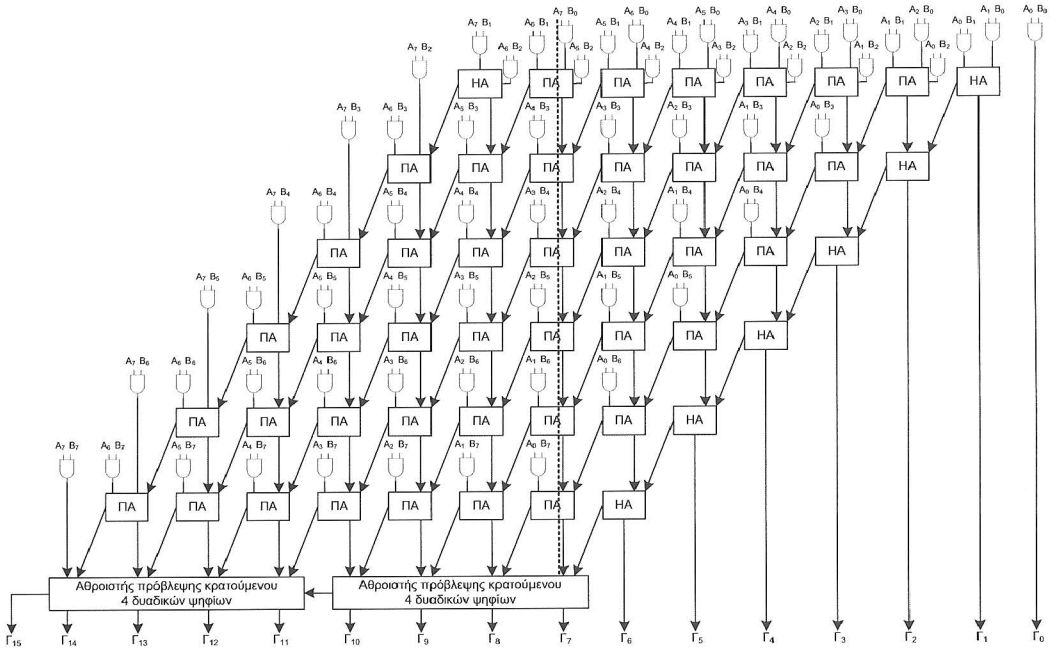
Για να υπολογίσουμε την καθυστέρηση των πολλαπλασιαστών θεωρούμε ότι η καθυστέρηση για τον υπολογισμό του κρατούμενου εξόδου και του αθροίσματος είναι 2 και 3 πύλες αντίστοιχα για τον ΠΑ και 1 και 2 πύλες αντίστοιχα για τον ΗΑ (βλέπε αθροιστές σελίδα 98, του βιβλίου Αρχιτεκτονική Υπολογιστών). Η καθυστέρηση του αθροιστή πρόβλεψης κρατούμενου (Carry Look-Ahead, CLA, adder) των 4 δυαδικών ψηφίων είναι ίση με την καθυστέρηση 5 λογικών πυλών.

Στο σχήμα 3.7.1 με έντονη διακεκομμένη γραμμή σημειώνεται ένα από τα μονοπάτια μέγιστης καθυστέρησης. Επομένως η μέγιστη καθυστέρηση $T_{\text{διατ.κ}}$ στον πολλαπλασιαστή διατήρησης κρατούμενου είναι:

$$T_{\text{διατ.κ}} = t_{\text{AND}} + 6 \times t_{\text{IIAsum}} + t_{\text{ATB}},$$

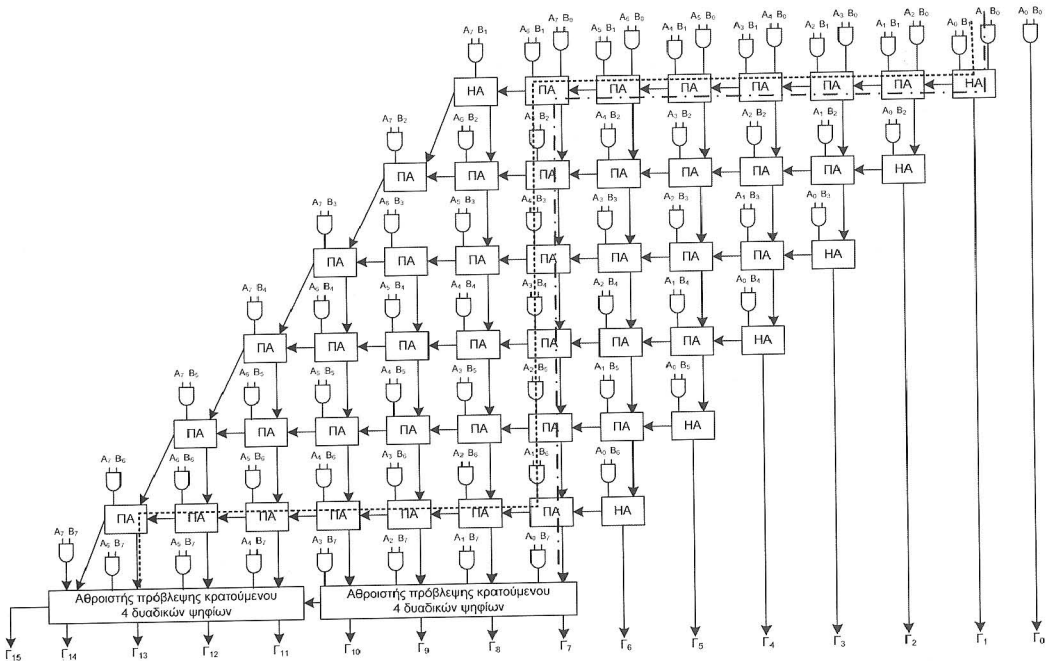
όπου t_{ATB} είναι η καθυστέρηση που βάζει ο αθροιστής της τελευταίας βαθμίδας.

Επομένως $T_{\text{διατ.κ}} = t_{\text{AND}} + 6 \times t_{\text{IIAsum}} + 2 \times t_{\text{CLA-4}}$, όπου $t_{\text{CLA-4}}$ είναι η καθυστέρηση του αθροιστή πρόβλεψης κρατούμενου των 4 δυαδικών ψηφίων σε επίπεδα πυλών, άρα $T_{\text{διατ.κ}} = 1 + 6 \times 3 + 2 \times 5 = 29$ επίπεδα πυλών.



Σχήμα 3.7.1 Πολλαπλασιαστής διατήρησης κρατούμενου για μη προσημασμένους αριθμούς των 8 δυαδικών ψηφίων, του οποίου η τελευταία βαθμίδα άθροισης υλοποιείται από δύο αθροιστές πρόβλεψης κρατούμενου των τεσσάρων δυαδικών ψηφίων ο κάθε ένας και διάδοσης κρατούμενου μεταξύ αυτών.

Για να υπολογίσουμε την καθυστέρηση του αντίστοιχου πολλαπλασιαστή διάδοσης κρατούμενου στο σχήμα 3.7.2 δίνουμε την δομή του.



Σχήμα 3.7.2 Πολλαπλασιαστής διάδοσης κρατούμενου για μη προσημασμένους αριθμούς των 8 δυαδικών ψηφίων, του οποίου η τελευταία βαθμίδα άθροισης υλοποιείται από δύο αθροιστές πρόβλεψης κρατούμενου των τεσσάρων δυαδικών ψηφίων ο κάθε ένας και διάδοσης κρατούμενου μεταξύ αυτών.

Στο σχήμα έχουν σημειωθεί δύο πιθανά μονοπάτια μέγιστης καθυστέρησης εκ των οποίων το ένα περνάει και από τους δύο αθροιστές πρόβλεψης κρατούμενου. Επειδή οι δύο αυτοί αθροιστές είναι συνδεδεμένοι στη σειρά (το κρατούμενο εξόδου του ενός οδηγεί το κρατούμενο εισόδου του άλλου) και η μέγιστη καθυστέρηση ενός αθροιστή πρόβλεψης κρατούμενου των 4 δυαδικών ψηφίων είναι 5 επίπεδα πυλών, η συνολική καθυστέρηση των δύο αθροιστών πρόβλεψης κρατούμενου θα είναι το πολύ ίση με $2 \times 5 = 10$ επίπεδα πυλών. Υπολογίζοντας την καθυστέρηση του υπομονοπατιού που περνάει παράλληλα πάνω από τους αθροιστές πρόβλεψης κρατούμενου έχουμε καθυστέρηση $6 \times t_{HA_{carry}} = 6 \times 2 = 12$ επίπεδα πυλών. Επομένως το μονοπάτι μέγιστης καθυστέρησης είναι αυτό που δεν περνάει από τον αθροιστή πρόβλεψης κρατούμενου των 4 δυαδικών ψηφίων που προσθέτει τα λιγότερο σημαντικά δυαδικά ψηφία.

$$T_{\text{διαδ.κ}} = t_{\text{AND}} + t_{\text{HAcarry}} + 11 \times t_{\text{IIAcarry}} + 6 \times t_{\text{IIAsum}} + 1 \times t_{\text{CLA-4}},$$

Άρα $T_{\text{διαδ.κ}} = 1 + 1 + 11 \times 2 + 6 \times 3 + 1 \times 5 = 47$ επίπεδα πυλών.

Παρατηρούμε ότι ο πολλαπλασιαστής διάδοσης κρατούμενου έχει πολύ μεγαλύτερη καθυστέρηση, σχεδόν διπλάσια.

Από το βιβλίο, Αρχιτεκτονική Υπολογιστών, σελίδα 114, έχουμε ότι η καθυστέρηση του αντίστοιχου πολλαπλασιαστή με αθροιστή διάδοσης κρατούμενου στην τελευταία βαθμίδα δίνεται από τη σχέση:

$$T_{\delta\kappa} = t_{\text{AND}} + t_{\text{HAcarry}} + 2 \times (v-2) \times t_{\text{IIAcarry}} + (v-1) \times t_{\text{IIAsum}}, \text{ οπότε}$$

$$T_{\delta\kappa} = 1 + 1 + 2 \times (8-2) \times 2 + (8-1) \times 3 = 47 \text{ επίπεδα πυλών.}$$

Παρατηρούμε ότι, στη συγκεκριμένη περίπτωση, χρησιμοποιώντας ένα πιο γρήγορο αθροιστή στην τελευταία βαθμίδα δεν κερδίζουμε σε ταχύτητα, ενώ χάνουμε σε υλικό (χρησιμοποιούμε περισσότερα κυκλώματα).

Άσκηση 3.8 (3.5 βιβλίου)

Να σχεδιάσετε πολλαπλασιαστή διατήρησης κρατούμενου των τεσσάρων δυαδικών ψηφίων που να μπορεί να χρησιμοποιηθεί είτε για πολλαπλασιασμό αριθμών χωρίς πρόσημο είτε για αριθμούς σε παράσταση συμπληρώματος ως προς δύο.

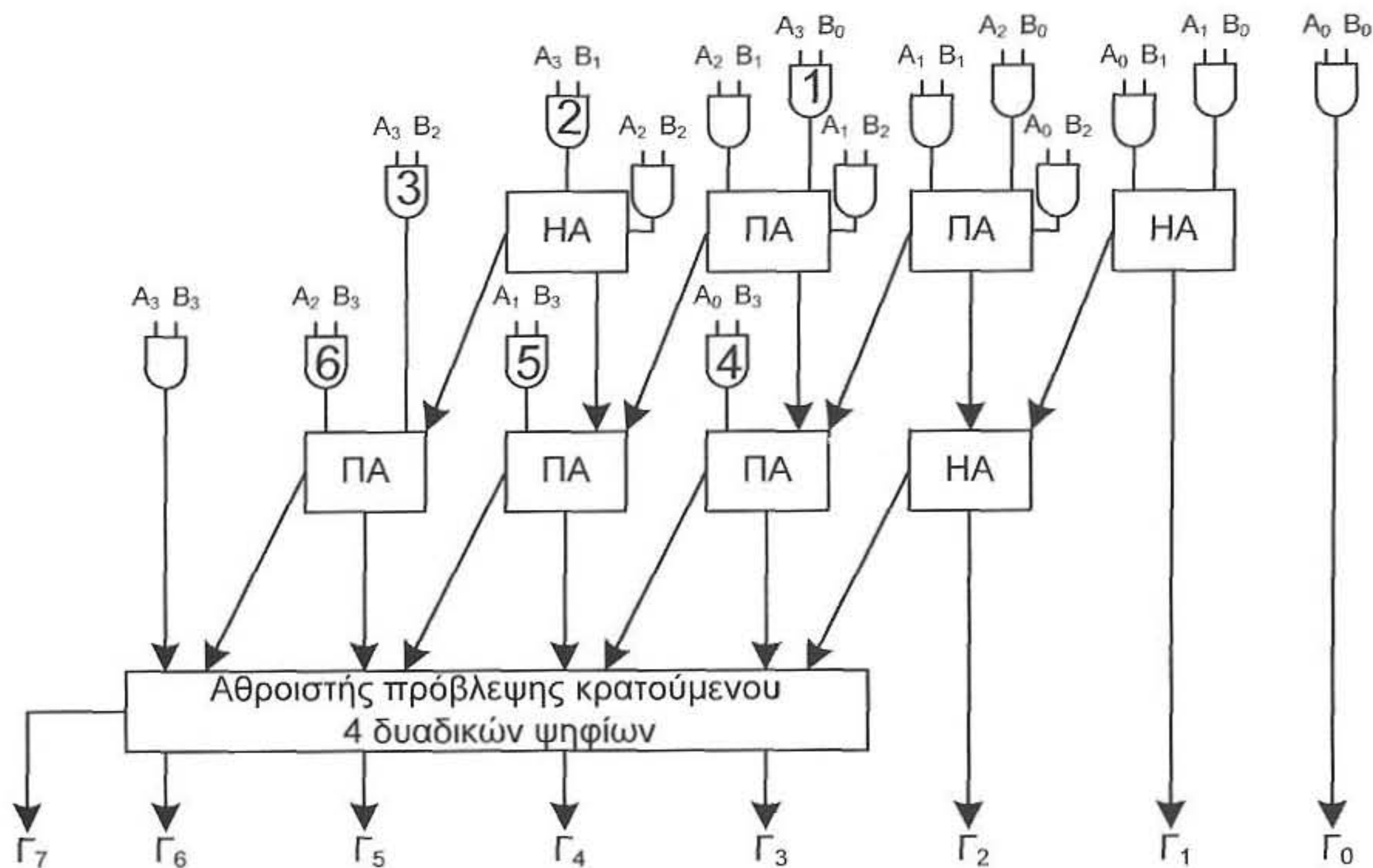
Απάντηση

Τα σχήματα 3.8.1 και 3.8.2 που ακολουθούν (αντιστοιχούν στα σχήματα 3.15 και 3.16 του βιβλίου Αρχιτεκτονική Υπολογιστών, Δημήτρης Νικολός, Εκδόσεις Β. Γκιούρδας Εκδοτική, Αθήνα 2008) δίνουν την δομή ενός πολλαπλασιαστή διατήρησης κρατούμενου των 4 δυαδικών ψηφίων μεταξύ αριθμών χωρίς πρόσημο και μεταξύ αριθμών σε παράσταση συμπληρώματος ως προς 2 αντίστοιχα. Παρατηρούμε ότι διαφέρουν στα ακόλουθα:

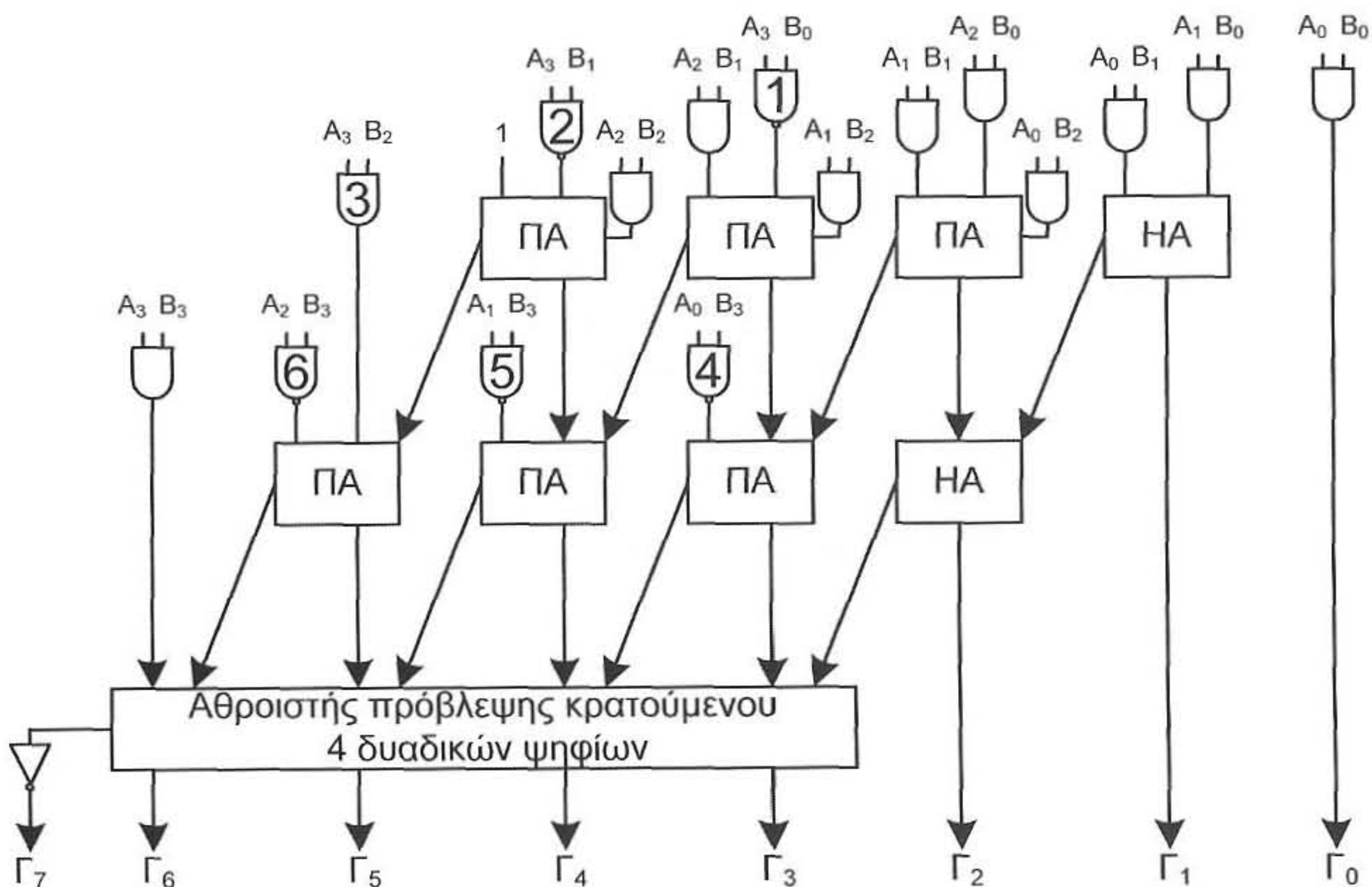
α. οι πύλες με αριθμούς 1, 2, 3, 4, 5 και 6 στο σχήμα 3.8.1 είναι πύλες AND ενώ στο σχήμα 3.8.2 είναι πύλες NAND

β. στην πιο σημαντική θέση του αθροιστή που κάνει την πρώτη άθροιση στο σχήμα 3.8.1 είναι ένας ημιαθροιστής, ΗΑ, ενώ στο σχήμα 3.8.2 είναι ένας πλήρης αθροιστής, ΠΑ, του οποίου η μία είσοδος οδηγείται με τη λογική τιμή 1.

γ. το πιο σημαντικό ψηφίο του γινομένου στον πολλαπλασιαστή του σχήματος 3.8.1 είναι το κρατούμενο εξόδου του αθροιστή της τελευταίας βαθμίδας, ενώ στο σχήμα 3.8.2 είναι το συμπλήρωμά του.



Σχήμα 3.8.1 Πολλαπλασιαστής διατήρησης κρατούμενου των τεσσάρων δυαδικών ψηφίων με αθροιστή πρόβλεψης κρατούμενου στην τελευταία βαθμίδα.



Σχήμα 3.8.2 Ένας πολλαπλασιαστής 4 δυαδικών ψηφίων διατήρησης κρατούμενου για αριθμούς σε αναπαράσταση συμπληρώματος ως προς δύο. Η πύλη NOT είναι ισοδύναμη με μια πύλη XOR της οποίας η μια είσοδος οδηγείται με τη λογική τιμή 1.

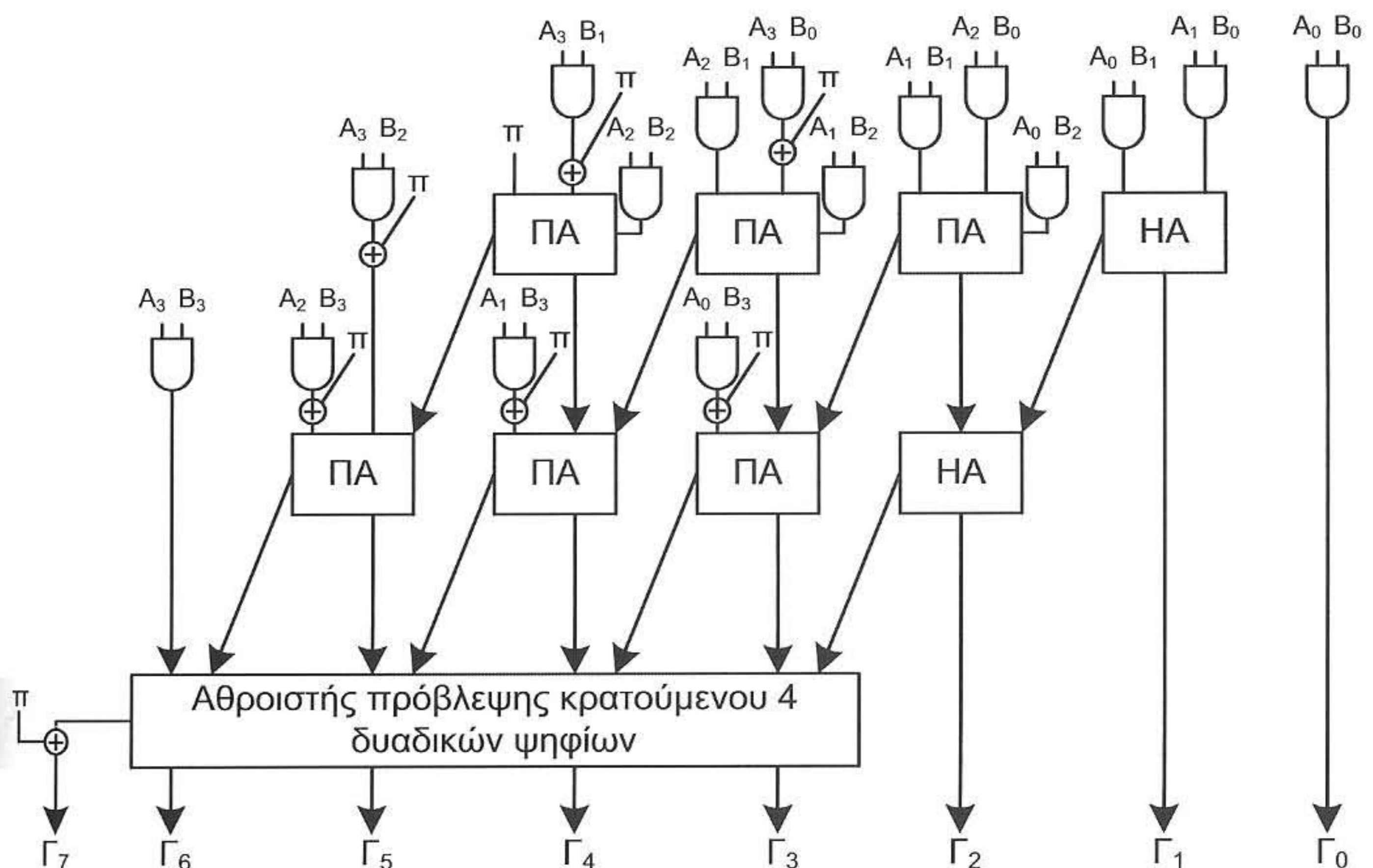
Για να σχεδιάσουμε ένα κύκλωμα με μία είσοδο ελέγχου π , ανάλογα με την τιμή της οποίας το κύκλωμα θα εκτελεί την πράξη του πολλαπλασιασμού μεταξύ αριθμών χωρίς πρόσημο ($\pi = 0$) ή μεταξύ αριθμών σε αναπαράσταση συμπληρώματος ως προς 2 ($\pi = 1$), θα κάνουμε τις ακόλουθες αλλαγές στο κύκλωμα του σχήματος 3.8.2:

α. Η πύλη NAND γνωρίζουμε ότι είναι ισοδύναμη με μία πύλη AND που οδηγεί μία πύλη NOT. Η πύλη NOT μπορεί να αντικατασταθεί με μία πύλη XOR της οποίας η άλλη είσοδος οδηγείται με το σήμα επιλογής π .

β. Η είσοδος του ΠΑ του αθροιστή που κάνει την πρώτη άθροιση και οδηγείται με τη λογική τιμή 1, θα οδηγείται από το σήμα ελέγχου π .

γ. Η πύλη NOT που οδηγείται από το κρατούμενο εξόδου του αθροιστή της τελευταίας βαθμίδας θα αντικατασταθεί από μία πύλη XOR της οποίας η μια είσοδος θα οδηγείται από το κρατούμενο εξόδου του αθροιστή αυτού και η άλλη είσοδος θα οδηγείται από το σήμα ελέγχου π .

Το κύκλωμα που θα προκύψει δίνεται στο σχήμα 3.8.3.



Σχήμα 3.8.3 Ένας πολλαπλασιαστής διάδοσης κρατούμενου των 4 δυαδικών ψηφίων για αριθμούς χωρίς πρόσημο ($\pi=0$) και αριθμούς σε αναπαράσταση συμπληρώματος ως προς 2 ($\pi=1$).

Άσκηση 3.9 (3.6 βιβλίου)

Θεωρήστε έναν μικροεπεξεργαστή με μονάδα ελέγχου που έχει σχεδιαστεί με την τεχνική του μικροπρογραμματισμού ενός επιπέδου. Το σύνολο των μικροπρογραμμάτων αποτελείται από 10.000 μικροεντολές και υπάρχουν 500 διαφορετικές μικροεντολές.

α) Ποιο είναι το πλεονέκτημα και ποιο το μειονέκτημα της οργάνωσης μικροπρογραμματισμένης μονάδας ελέγχου δύο επιπέδων; Μπορείτε να χρησιμοποιήσετε αυτή την τεχνική στον παραπάνω μικροεπεξεργαστή;

β) Καθορίστε τον ελάχιστο αριθμό των δυαδικών ψηφίων που πρέπει να έχει η μικροεντολή, προκειμένου με την οργάνωση δύο επιπέδων να επιτυγχάνουμε μείωση του απαιτούμενου αποθηκευτικού χώρου κατά τουλάχιστον 60%.

Απάντηση

α. Όταν στα μικροπρογράμματα εμφανίζονται συχνά όμοιες μικροεντολές χρησιμοποιώντας οργάνωση δύο επιπέδων μπορούμε να επιτύχουμε μείωση της χωρητικότητας της μνήμης που απαιτείται για την υλοποίηση της μνήμης ελέγχου. Στην οργάνωση δύο επιπέδων στο δεύτερο επίπεδο (νανομνήμη) αποθηκεύονται όλες οι διαφορετικές μικροεντολές (νανοεντολές), ενώ στο πρώτο επίπεδο αποθηκεύονται τα μικροπρογράμματα που τώρα αποτελούνται από διευθύνσεις της νανομνήμης.

Στην περίπτωση μας τα μικροπρογράμματα καταλαμβάνουν χώρο 10.000 μικροεντολών και απαρτίζονται μόνο από 500 διαφορετικές μικροεντολές. Άρα κατά μέσο όρο κάθε μικροεντολή επαναλαμβάνεται $10.000/500 = 20$ φορές, αριθμός αρκετά μεγάλος για να χρησιμοποιηθεί η τεχνική μνήμης ελέγχου δύο επιπέδων.

Το μειονέκτημα της οργάνωσης δύο επιπέδων είναι ότι απαιτούνται δύο προσπελάσεις, μία στη μνήμη του πρώτου επιπέδου και μία στη μνήμη του δεύτερου. Αντίθετα με τη χρήση ενός επιπέδου απαιτείται μόνο μία προσπέλαση, επομένως είναι ταχύτερη η εκτέλεση κάθε μικροεντολής.

β. Έστω ότι το μήκος της κάθε μικροεντολής είναι μ . Τότε για την υλοποίηση της μνήμης ελέγχου ενός επιπέδου απαιτούνται $10.000 \times \mu$ δυαδικά ψηφία.

Όταν η μνήμη ελέγχου υλοποιείται με την τεχνική του νανοπρογραμματισμού, τότε το δεύτερο επίπεδο περιέχει μόνο μία φορά την κάθε μικροεντολή, επομένως αποτελείται από $500 \times \mu$ δυαδικά ψηφία, όσα ακριβώς απαιτούνται για την αποθήκευση των 500 διαφορετικών μικροεντολών. Το πρώτο επίπεδο έχει 10.000 θέσεις, όσες απαιτούνται για τα μικροπρόγραμμα στην περίπτωση του ενός επιπέδου, μόνο που τώρα το μήκος κάθε θέσης είναι διαφορετικό, αφού περιέχει τη διεύθυνση μιας θέσης της νανομνήμης.

Αφού λοιπόν η νανομνήμη έχει 500 θέσεις, τότε κάθε θέση μνήμης του πρώτου επιπέδου θα έχει μήκος $v = \lceil \log_2(500) \rceil = 9$ δυαδικά ψηφία. Επομένως στην περίπτωση του νανοπρογραμματισμού απαιτούνται $10.000 \times 9 + 500 \times \mu$ δυαδικά ψηφία.

Ο λόγος μείωσης ΛΜ του απαιτούμενου αποθηκευτικού χώρου για τις δύο τεχνικές είναι $\Lambda M = [10.000 \times \mu - (10.000 \times 9 + 500 \times \mu)] / (10.000 \times \mu)$ επομένως για να έχουμε $\Lambda M \geq 60\%$ πρέπει $[10.000 \times \mu - (10.000 \times 9 + 500 \times \mu)] / (10.000 \times \mu) \geq 0,6$ ή ισοδύναμα

$$10.000 \times \mu - 90.000 - 500 \times \mu \geq 6000 \times \mu \quad \text{ή} \quad 10.000 \times \mu - 6000 \times \mu - 500 \times \mu \geq 90.000 \quad \text{ή} \\ 3500 \times \mu \geq 90.000 \quad \text{ή} \quad \mu \geq 90.000 / 3500 = 25,7 \quad \text{συνεπώς} \quad \mu \geq 26$$

Άσκηση 3.10 (3.7 βιβλίου)

Θεωρήστε ότι το σύνολο των μικροεντολών που είναι αποθηκευμένο στη μνήμη ελέγχου μίας μικροπρογραμματισμένης μονάδας ελέγχου είναι:

Μικροεντολή	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
μ_0	1	1	0	0	1	0	0	0	0	0	0
μ_1	0	1	0	0	0	0	1	0	0	0	0
μ_2	0	0	1	0	0	1	0	0	0	0	0
μ_3	0	0	0	0	1	1	0	0	0	1	0
μ_4	0	0	0	0	1	0	0	0	1	0	0
μ_5	1	0	1	1	0	0	0	0	0	0	0
μ_6	0	0	0	1	0	0	0	1	0	0	0
μ_7	0	0	0	0	0	0	0	0	0	1	1

(Προφανώς το σύνολο των μικροπρογραμμάτων μίας μικροπρογραμματισμένης μονάδας ελέγχου δεν αποτελείται ποτέ από τόσο μικρό αριθμό μικροεντολών, ούτε και έχει τόσο μικρό αριθμό σημάτων ελέγχου. Στην άσκηση επιλέξαμε τόσο μικρό πλήθος μικροεντολών και σημάτων ελέγχου για να μπορούμε εύκολα να τα χειριστούμε).

Θεωρήστε ότι κάθε σήμα ελέγχου ενεργοποιεί μία μικρολειτουργία.

α. Να εφαρμόσετε την τεχνική της κωδικοποίησης των σημάτων ελέγχου για να ελαττωθεί η χωρητικότητα της απαιτούμενης μνήμης ελέγχου χωρίς να μειωθεί το πλήθος των μικρολειτουργιών που μπορούν να εκτελεστούν κάθε χρονική περίοδο παράλληλα.

β. Να υπολογίσετε την απαιτούμενη χωρητικότητα της μνήμης ελέγχου που προκύπτει με την εφαρμογή της κωδικοποίησης των σημάτων ελέγχου και το ποσοστό μείωσης της απαιτούμενης χωρητικότητας.

γ. Να δώσετε τα νέα περιεχόμενα της μνήμης ελέγχου.

Απάντηση

α. Αφού έχουμε θεωρήσει ότι κάθε σήμα ελέγχου ενεργοποιεί μία μικρολειτουργία, σε κάθε χρονική περίοδο θα εκτελούνται τόσες μικρολειτουργίες όσες μονάδες υπάρχουν στην μικροεντολή που εκτελείται αυτή την περίοδο. Σύμφωνα με την τεχνική της κωδικοποίησης των σημάτων ελέγχου για να μην ελαττωθεί το πλήθος των μικρολειτουργιών που εκτελούνται σε κάθε χρονική περίοδο θα πρέπει να μην κωδικοποιηθούν μαζί σήματα ελέγχου που είναι ενεργά (έχουν την τιμή 1) στην ίδια μικροεντολή. Επομένως θα πρέπει να χωρίσουμε τα σήματα ελέγχου σε ομάδες έτσι ώστε κατά την διάρκεια του ίδιου κύκλου ρολογιού σε κάθε ομάδα να μην παίρνουν την τιμή 1 περισσότερα από ένα σήματα ελέγχου.

Μία από τις επιτρεπτές διαμερίσεις σε ομάδες είναι η ακόλουθη:

$\{C_0, C_5, C_{10}\}, \{C_1, C_2, C_7, C_8, C_9\}, \{C_3, C_4, C_6\}$

β. Για την ομάδα σημάτων $\{C_0, C_5, C_{10}\}$ απαιτείται η αποθήκευση δύο τιμών (M_0 και M_1) σε κάθε μικροεντολή, για την ομάδα $\{C_1, C_2, C_7, C_8, C_9\}$ η αποθήκευση τριών τιμών (M_2, M_3 και M_4) σε κάθε μικροεντολή και για την ομάδα $\{C_3, C_4, C_6\}$ απαιτείται η αποθήκευση δύο τιμών (M_5 και M_6) σε κάθε μικροεντολή.

Επομένως η χωρητικότητα της μνήμης ελέγχου θα είναι:

8 μικροεντολές \times (2+3+2)δυαδικά ψηφία ανά μικροεντολή = 56 δυαδικά ψηφία.

Η μείωση που επιτυγχάνεται είναι $[(8 \times 11 - 8 \times 7) / (8 \times 10)] \times 100 = [(88 - 56) / 80] \times 100 = 40\%$.

γ. Έστω:

$M_0 \ M_1$

0 0 \rightarrow δεν ενεργοποιείται κανένα από τα σήματα C_0, C_5 ή C_{10}

1 0 \rightarrow ενεργοποίηση του σήματος C_0

0 1 \rightarrow ενεργοποίηση του σήματος C_5

1 1 \rightarrow ενεργοποίηση του σήματος C_{10}

$M_2 \ M_3 \ M_4$

0 0 0 \rightarrow δεν ενεργοποιείται κανένα από τα σήματα C_1, C_2, C_7, C_8 ή C_9

1 0 0 \rightarrow ενεργοποίηση του σήματος C_1

0 1 0 \rightarrow ενεργοποίηση του σήματος C_2

- $1 \ 1 \ 0 \rightarrow$ ενεργοποίηση του σήματος C_7
 $0 \ 0 \ 1 \rightarrow$ ενεργοποίηση του σήματος C_8
 $1 \ 0 \ 1 \rightarrow$ ενεργοποίηση του σήματος C_9
 $0 \ 1 \ 1 \rightarrow$ δεν ενεργοποιείται κανένα από τα σήματα C_1, C_2, C_7, C_8 ή C_9
 $1 \ 1 \ 1 \rightarrow$ δεν ενεργοποιείται κανένα από τα σήματα C_1, C_2, C_7, C_8 ή C_9

$M_5 \ M_6$

- $0 \ 0 \rightarrow$ δεν ενεργοποιείται κανένα από τα σήματα C_3, C_4 ή C_6
 $1 \ 0 \rightarrow$ ενεργοποίηση του σήματος C_3
 $0 \ 1 \rightarrow$ ενεργοποίηση του σήματος C_4
 $1 \ 1 \rightarrow$ ενεργοποίηση του σήματος C_6

Τότε στη μνήμη ελέγχου θα είναι αποθηκευμένη η ακόλουθη πληροφορία:

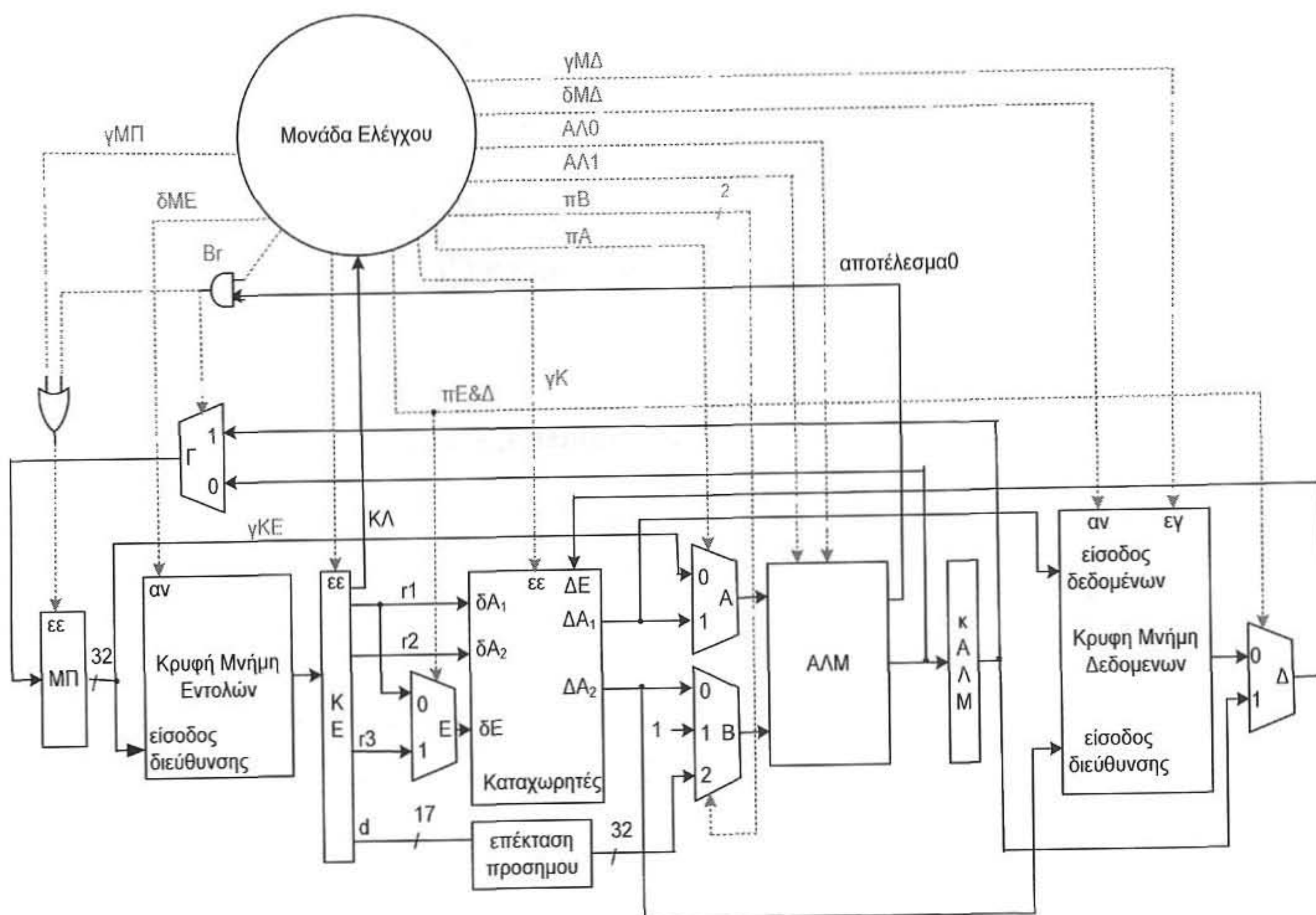
Μικροεντολή	M_0	M_1	M_2	M_3	M_4	M_5	M_6
μ_0	1	0	1	0	0	0	1
μ_1	0	0	1	0	0	1	1
μ_2	0	1	0	1	0	0	0
μ_3	0	1	1	0	1	0	1
μ_4	0	0	0	0	1	0	1
μ_5	1	0	0	1	0	1	0
μ_6	0	0	1	1	0	1	0
μ_7	1	1	1	0	1	0	0

Άσκηση 3.11

Η μονάδα επεξεργασίας δεδομένων σταθερής υποδιαστολής του σχήματος 3.11.1 (σχήμα 3.60 του βιβλίου) έχει σχεδιαστεί για την εκτέλεση των εντολών: $LOAD \ r1, (r2)$ [$r1 \leftarrow M(r2)$], $STORE \ r1, (r2)$ [$r1 \rightarrow M(r2)$], $ADD \ r1, r2, r3$ [$r1+r2 \rightarrow r3$], $SUB \ r1, r2, r3$ [$r1-r2 \rightarrow r3$], $AND \ r1, r2, r3$ [$r1 \wedge r2 \rightarrow r3$], και $BRE \ r1, r2, d$ [εάν $r1-r2=0$ τότε $ΜΠ=ΜΠ+d$], όπου στις αγκύλες είναι η επεξήγηση της εντολής.

Να κάνετε τις απαιτούμενες τροποποιήσεις ώστε η μονάδα να μπορεί να εκτελέσει:

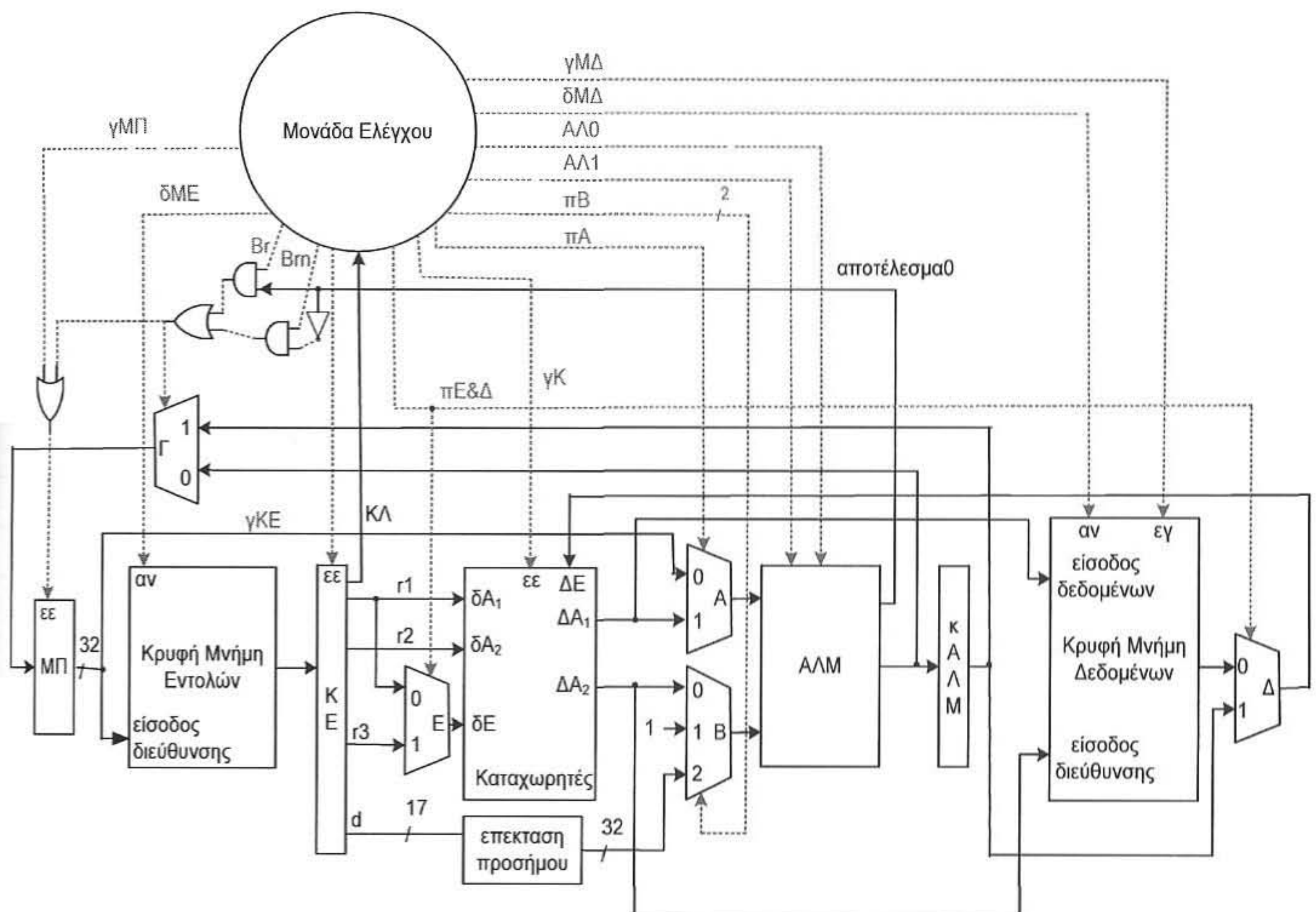
- την εντολή $BRNE \ r1, r2, d$ [εάν $r1-r2 \neq 0$ τότε $ΜΠ=ΜΠ+d$].
- την εντολή $BRM \ r1, r2, d$ [εάν $r1-r2 < 0$ τότε $ΜΠ=ΜΠ+d$].



Σχήμα 3.11.1

Απάντηση

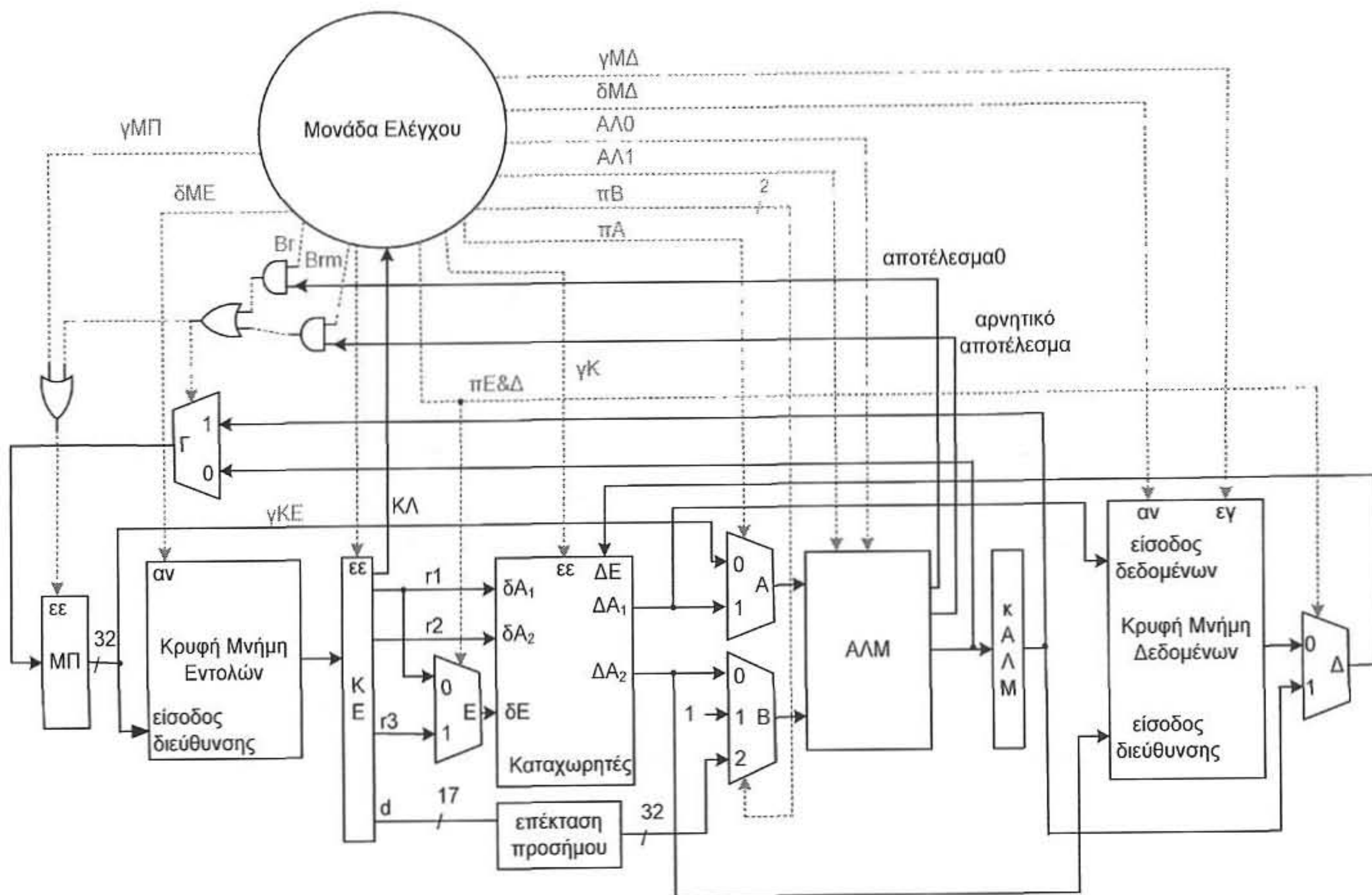
α. Κατά την εκτέλεση μιας εντολής «BRE r1, r2, d» ή «BRNE r1, r2, d» κατά τον τρίτο κύκλο ρολογιού γίνεται η αφαίρεση $r1-r2$ και στο τέλος αυτού του κύκλου, κατά την αλλαγή του χρονικού σήματος από 0 σε 1, ανάλογα του αποτελέσματος γίνεται η ενημέρωση ή όχι του μετρητή προγράμματος, ΜΠ. Επομένως η μονάδα ελέγχου πρέπει να παράγει ένα επιπλέον σήμα ελέγχου Brn , το οποίο όταν η εκτελούμενη εντολή είναι η εντολή «BRNE r1, r2, d» κατά τη διάρκεια του τρίτου κύκλου θα παίρνει την ενεργό τιμή (λογική τιμή ένα). Στο σχήμα 3.11.2 δίνονται οι απαιτούμενες αλλαγές για να μπορεί να εκτελείται και η εντολή «BRNE r1, r2, d».



Σχήμα 3.11.2 Το σήμα Br είναι ενεργό (έχει τη λογική τιμή 1) κατά τη διάρκεια του τρίτου κύκλου εκτέλεσης μιας εντολής «BRE r1, r2 d», ενώ το σήμα Brn είναι ενεργό κατά τον ίδιο κύκλο εκτέλεσης μιας εντολής «BRNE r1, r2, d».

β.

Κατά την εκτέλεση μιας εντολής «BRE r1, r2, d» ή «BRM r1, r2, d» κατά τον τρίτο κύκλο ρολογιού γίνεται η αφαίρεση $r1 - r2$ και στο τέλος του τρίτου κύκλου, κατά την αλλαγή του χρονικού σήματος από 0 σε 1, ανάλογα του αποτελέσματος γίνεται η ενημέρωση ή όχι του μετρητή προγράμματος, ΜΠ. Επομένως η μονάδα ελέγχου πρέπει να παράγει ένα επιπλέον σήμα ελέγχου Brm, το οποίο όταν η εκτελούμενη εντολή είναι η εντολή «BRM r1, r2, d» κατά τη διάρκεια του τρίτου κύκλου θα παίρνει την ενεργό τιμή (λογική τιμή ένα). Στο σχήμα 3.11.3 δίνονται οι απαιτούμενες αλλαγές για να μπορεί να εκτελείται και η εντολή «BRM r1, r2, d».



Σχήμα 3.11.3 Το σήμα Br είναι ενεργό (έχει τη λογική τιμή 1) κατά τη διάρκεια του τρίτου κύκλου εκτέλεσης μιας εντολής «BRE r1, r2 d», ενώ το σήμα Brm είναι ενεργό κατά τον ίδιο κύκλο εκτέλεσης μιας εντολής «BRM r1, r2, d».

Στο σχήμα 3.11.3 η γραμμή αρνητικό αποτέλεσμα λαμβάνει την τιμή 1 όταν το αποτέλεσμα της πράξης στην ΑΛΜ είναι αρνητικό. Δηλαδή όταν δεν υπάρχει υπερχείλιση και το πιο σημαντικό δυαδικό ψηφίο του αποτελέσματος έχει τη λογική τιμή 1 ή όταν υπάρχει υπερχείλιση και το πιο σημαντικό δυαδικό ψηφίο του αποτελέσματος έχει τη λογική τιμή 0 ($Y' \cdot \Pi\Sigma\Psi + Y \cdot \Pi\Sigma\Psi'$).

Άσκηση 3.12 (3.8 βιβλίου)

Θεωρούμε ένα υπολογιστή του οποίου όλες οι εντολές και τα δεδομένα σταθερής υποδιαστολής είναι των 32 δυαδικών ψηφίων. Θεωρούμε ότι η κύρια μνήμη είναι οργανωμένη έτσι ώστε κάθε θέση της να είναι των 32 δυαδικών ψηφίων και ότι ο υπολογιστής μας διαθέτει ξεχωριστή κρυφή μνήμη εντολών και ξεχωριστή κρυφή μνήμη δεδομένων και ότι όλες οι εντολές του προγράμματος βρίσκονται στην κρυφή μνήμη εντολών και όλα τα δεδομένα στην κρυφή μνήμη δεδομένων. Θεωρούμε επίσης ότι η μονάδα επεξεργασίας δεδομένων περιλαμβάνει 16 καταχωρητές γενικού σκοπού με δύο πόρτες ανάγνωσης και μία πόρτα εγγραφής.

Να σχεδιάσετε μονάδα επεξεργασίας δεδομένων σταθερής υποδιαστολής για το υποσύνολο εντολών του Πίνακα 3.12.1.

Πίνακας 3.12.1

Εντολή	Ερμηνεία
LOAD r1, (r2)	$r1 \leftarrow M(r2)$
STORE r1, (r2)	$M(r2) \leftarrow r1$
ADD r1, r2, r3	$r3 \leftarrow r1 + r2$
SUB r1, r2, r3	$r3 \leftarrow r1 - r2$
BRZ d	εάν $\sigma M = 1$ τότε $ΜΠ = ΜΠ + d$
BRC d	εάν $\sigma K = 1$ τότε $ΜΠ = ΜΠ + d$

$M(r2)$: έμμεσος τρόπος διευθυνσιοδότησης με χρήση του καταχωρητή r2

σM : σημαία μηδενικού αποτελέσματος (Zero flag)

σK : σημαία κρατούμενου εξόδου (Carry flag)

ΜΠ: Μετρητής Προγράμματος

Απάντηση

Θα σχεδιάσουμε μονάδα επεξεργασίας δεδομένων σταθερής υποδιαστολής που θα εκτελεί κάθε εντολή σε περισσότερους από ένα κύκλους. Αυτή η υλοποίηση όπως έχουμε δει στην ενότητα 3.3 του βιβλίου Αρχιτεκτονική Υπολογιστών έχει πλεονεκτήματα σε σχέση με την υλοποίηση που εκτελεί κάθε εντολή σε ένα κύκλο ρολογιού.

Οι εντολές «LOAD r1, (r2)», «STORE r1, (r2)», «ADD r1, r2, r3» και «SUB r1, r2, r3», είναι ίδιες με τις εντολές που θεωρήσαμε στην υποενότητα 3.3 του βιβλίου. Ωστόσο οι εντολές αλλαγής της ροής του προγράμματος «BRZ d» και «BRC d» είναι διαφορετικές. Τώρα η εντολή «BRZ d» θα εκτελεστεί εάν η σημαία μηδενικού αποτελέσματος του καταχωρητή κατάστασης (status register) έχει την τιμή ένα. Ομοίως η εντολή «BRC d» θα εκτελεστεί εάν η σημαία κρατούμενου εξόδου του καταχωρητή κατάστασης έχει την τιμή ένα. Από τις εντολές που μελετάμε, μόνο οι εντολές «ADD r1, r2, r3» και «SUB r1, r2, r3» θέτουν ανάλογα με το αποτέλεσμα τις σημαίες μηδενικού αποτελέσματος και κρατούμενου εξόδου του καταχωρητή κατάστασης.

Από το σύνολο εντολών που μελετάμε αντιλαμβανόμαστε ότι επιπλέον του μετρητή προγράμματος, της Κρυφής Μνήμης Εντολών, της Κρυφής Μνήμης Δεδομένων και των καταχωρητών γενικού σκοπού, θα πρέπει να υπάρχουν τουλάχιστον και μια Αριθμητική Λογική Μονάδα και ο καταχωρητής κατάστασης, κΚ. Η Αριθμητική Λογική Μονάδα θα έχει δύο εξόδους, την έξοδο μηδενικού αποτελέσματος και την έξοδο κρατούμενου εξόδου οι οποίες θα ενημερώνουν τις αντίστοιχες σημαίες του καταχωρητή κατάστασης.

Θεωρούμε ότι η μορφή των εντολών «LOAD r1, (r2)», «STORE r1, (r2)», «ADD r1, r2, r3» και «SUB r1, r2, r3» σε επίπεδο γλώσσας μηχανής είναι αυτή που θεωρήσαμε στα σχήματα 3.34, 3.38 και 3.40 του βιβλίου που αναφέραμε πιο πάνω.

Σε επίπεδο γλώσσας μηχανής οι εντολές διακλάδωσης BRZ και BRC έχουν την ίδια μορφή, η οποία δίνεται στο σχήμα 3.12.1, με μόνη διαφορά ότι έχουν διαφορετικούς κωδικούς λειτουργίας. Έστω ότι ο κωδικός λειτουργίας της εντολής BRZ είναι 1110000 ενώ της εντολής BRC είναι 1110001.

ΠΣΨ					ΛΣΨ
31	-	25	24	-	0
ΚΛ		d			

Κωδικός λειτουργίας, ΚΛ, 7 δυαδικών ψηφίων

Αριθμός μετατόπισης (displacement)
25 δυαδικά ψηφία

BRZ: ΚΛ=1 1 1 0 0 0 0, BRC: ΚΛ=1 1 1 0 0 0 1

Σχήμα 3.12.1 Μορφή εντολής “BRZ d” και “BRC d” σε επίπεδο γλώσσας μηχανής.

Θεωρούμε ότι κάθε μία από τις λειτουργίες: ανάγνωση της Κρυφής Μνήμης Εντολών, ανάγνωση ή αποθήκευση στους καταχωρητές γενικού σκοπού, αριθμητική ή λογική πράξη στην Αριθμητική Λογική Μονάδα και ανάγνωση ή εγγραφή στην Κρυφή Μνήμη Δεδομένων γίνεται σε ένα κύκλο ρολογιού.

Για τους καταχωρητές γενικού σκοπού ισχύει ότι αναφέραμε στην ενότητα 3.3 του βιβλίου που αναφέραμε.

Όπως είδαμε στην υποενότητα 2.3.2 του βιβλίου που αναφέραμε, οι δύο πρώτοι κύκλοι ρολογιού είναι κοινοί για όλες τις εντολές. Θα περιγράψουμε λοιπόν τι γίνεται στους δύο πρώτους κύκλους ρολογιού και στη συνέχεια ανάλογα με την εκτελούμενη εντολή τι γίνεται σε κάθε ένα από τους επόμενους κύκλους ρολογιού.

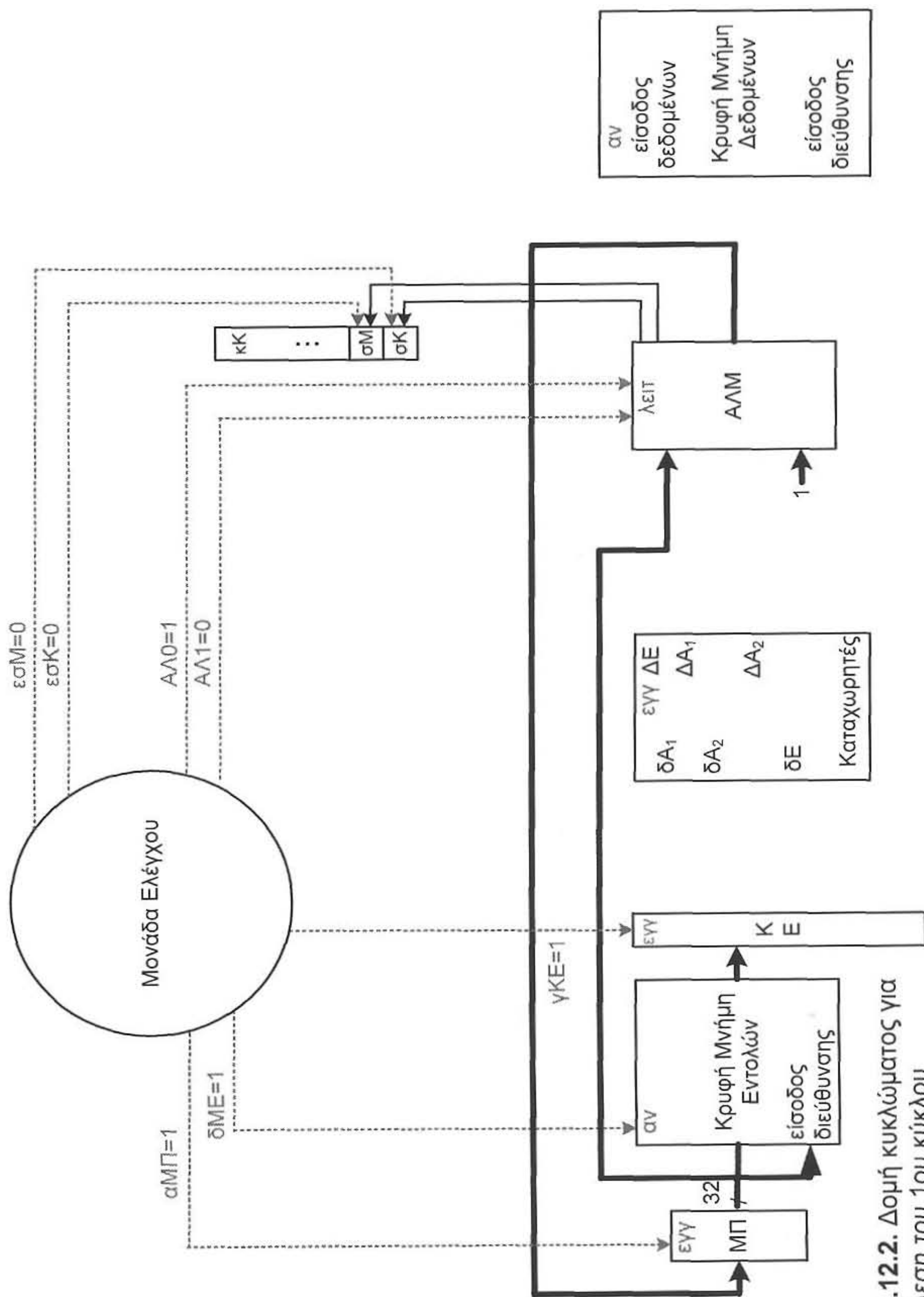
1^{ος} κύκλος ρολογιού.

Κατά τον πρώτο κύκλο ρολογιού γίνεται η προσκόμιση από την κρυφή μνήμη της επόμενης προς εκτέλεση εντολής, δηλαδή της εντολής που είναι αποθηκευμένη στη θέση μνήμης η διεύθυνση της οποίας υπάρχει στο Μετρητή Προγράμματος, ΜΠ. Γι' αυτό το λόγο η Μονάδα Ελέγχου παράγει ένα σήμα ελέγχου, το δΜΕ, το οποίο οδηγεί την είσοδο ενεργοποίησης ανάγνωσης της κρυφής μνήμης εντολών στη λογική τιμή 1.

Στον πρώτο κύκλο γίνεται επίσης η αύξηση του περιεχομένου του ΜΠ κατά ένα ώστε μετά την προσκόμιση της τρέχουσας εντολής το περιεχόμενό του να περιέχει την διεύθυνση της επόμενης εντολής του προγράμματος (αγνοούμε προς το παρόν την ύπαρξη εντολών διακλάδωσης). Ο ΜΠ αυξάνει το περιεχόμενό του κατά ένα διότι θεωρήσαμε ότι κάθε εντολή καταλαμβάνει μία θέση μνήμης. Αφού κατά τον πρώτο κύκλο ρολογιού δεν χρησιμοποιείται η διαθέσιμη ΑΛΜ για κάποιο άλλο σκοπό, θα την χρησιμοποιήσουμε για την αύξηση του περιεχομένου του ΜΠ κατά ένα. Ως εκ τούτου το περιεχόμενο του ΜΠ θα πρέπει να οδηγηθεί στην ΑΛΜ. Με το περιεχόμενο του ΜΠ θα πρέπει να προστεθεί στη ΑΛΜ μία μονάδα η οποία όπως βλέπουμε στο σχήμα 3.12.2 είναι είσοδος στην ΑΛΜ. Επομένως, κατά τον πρώτο κύκλο ρολογιού, οι εισόδοι δεδομένων της ΑΛΜ οδηγούνται από το περιεχόμενο του ΜΠ και τον αριθμό 1. Για να γίνει πρόσθεση οι εισόδοι ΑΛ1 και ΑΛ0 της ΑΛΜ οδηγούνται από την μονάδα ελέγχου στο 0 και 1 αντίστοιχα.

Κατά την διάρκεια του πρώτου κύκλου ρολογιού, άσχετα με το αποτέλεσμα της πρόσθεσης στην ΑΛΜ, οι σημαίες σΜ και σΚ δεν πρέπει να ενημερώνονται. Για το σκοπό αυτό τα σήματα ενεργοποίησης ενημέρωσης εσΜ και εσΚ των τιμών των σημαιών σΜ και σΚ αντίστοιχα έχουν την τιμή 0, οπότε οι σημαίες σΜ και σΚ διατηρούν τις τιμές που είχαν. Κατά την διάρκεια του πρώτου κύκλου ρολογιού το σήμα ελέγχου αΜΠ (αύξηση της τιμής του ΜΠ) έχει την τιμή 1, οπότε στο τέλος του πρώτου κύκλου ρολογιού, κατά την μετάβαση του ρολογιού από 0 σε 1, στον ΜΠ αποθηκεύονται τα δεδομένα εξόδου της ΑΛΜ που είναι η προηγούμενη τιμή του ΜΠ αυξημένη κατά ένα. Η ΑΛΜ σε κάποιο από τους επόμενους κύκλους ρολογιού πιθανόν να χρησιμοποιείται για κάποια άλλη πράξη. Επομένως ο ΜΠ δεν πρέπει να ενημερώνεται κάθε φορά που έχουμε μετάβαση του χρονικού σήματος από 0 σε 1. Αυτός είναι ο λόγος για τον οποίο ο ΜΠ έχει την είσοδο ενεργοποίησης εγγραφής αΜΠ.

Η διάρκεια του κύκλου ρολογιού έχει επιλεγεί να είναι μεγαλύτερη του χρόνου προσπέλασης της κρυφής μνήμης εντολών. Η είσοδος ενεργοποίησης εγγραφής του καταχωρητή εντολών ΚΕ οδηγείται από το σήμα ελέγχου γΚΕ. Το σήμα γΚΕ κατά τη διάρκεια του πρώτου κύκλου ρολογιού έχει την τιμή ένα ώστε κατά την μετάβαση του σήματος χρονισμού από μηδέν σε ένα στον καταχωρητή ΚΕ να αποθηκευτεί η προς εκτέλεση εντολή.



Σχήμα 3.12.2. Δομή κυκλώματος για την εκτέλεση του 1ου κύκλου ρολογιού

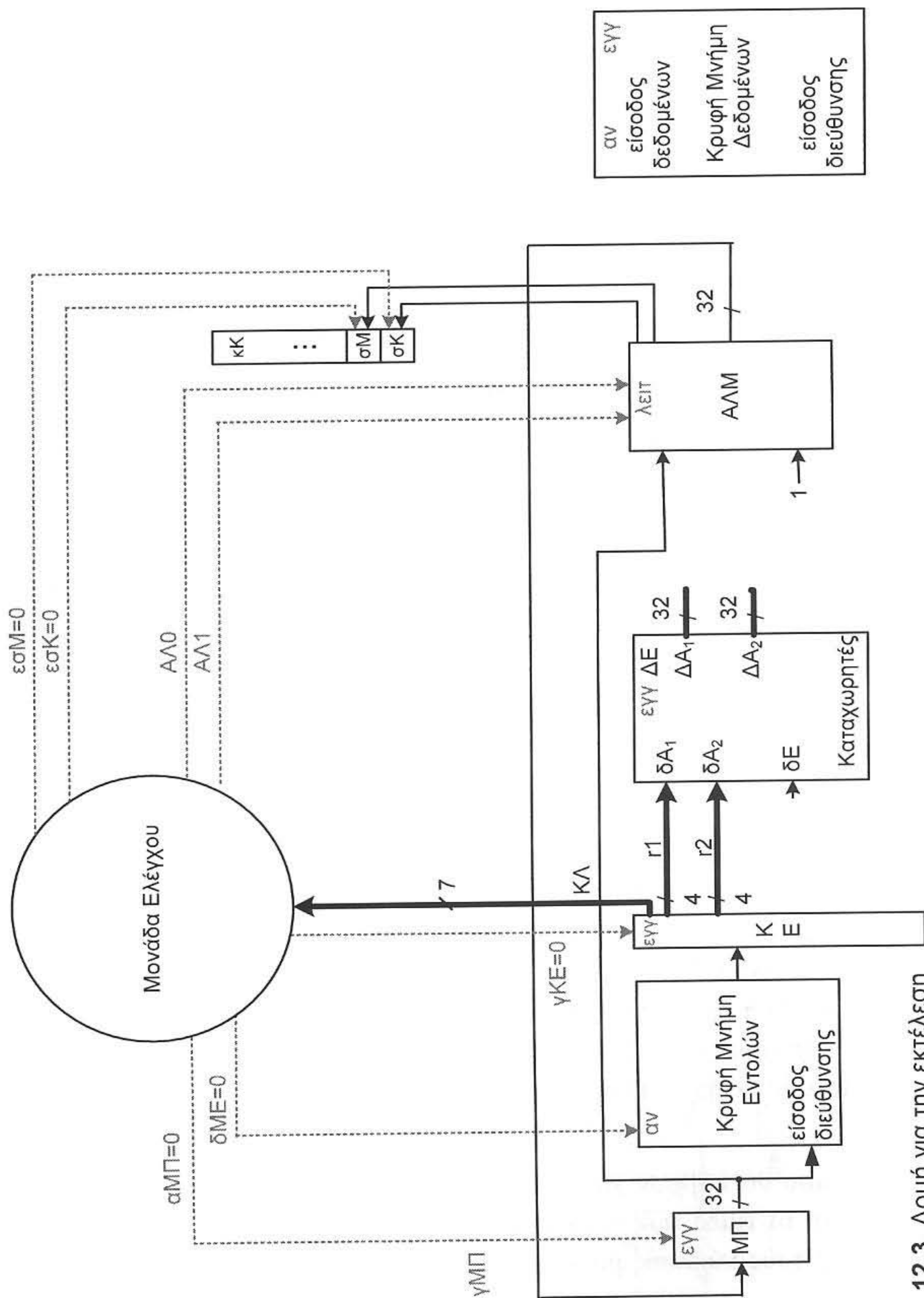
Πρέπει να σημειώσουμε ότι ο λόγος για τον οποίο ο καταχωρητής ΚΕ έχει είσοδο ενεργοποίησης εγγραφής είναι ότι, αφού κάθε εντολή εκτελείται σε περισσότερους από ένα κύκλους, θα πρέπει στον ΚΕ να είναι αποθηκευμένη η εκτελούμενη εντολή κατά την διάρκεια όλων των κύκλων ρολογιού μέχρι την ολοκλήρωση της εκτέλεσης της εντολής. Άρα ο ΚΕ δεν θα πρέπει να ενημερώνεται στο τέλος κάθε κύκλου ρολογιού. Επομένως σε όλους τους άλλους κύκλους το σήμα ελέγχου γΚΕ θα έχει την τιμή μηδέν.

Στο σχήμα 3.12.2 δίνονται οι τιμές των σημάτων ελέγχου κατά την διάρκεια του πρώτου κύκλου ρολογιού καθώς και η ροή της πληροφορίας με γραμμές μεγαλύτερου εύρους.

2^{ος} κύκλος ρολογιού.

Σύμφωνα με την ανάλυση που έγινε, με το ξεκίνημα του δευτέρου κύκλου ρολογιού, ο ΜΠ περιέχει την διεύθυνση της επομένης εντολής του προγράμματος και ο καταχωρητής εντολής ΚΕ την εντολή που θα εκτελεστεί. Το τμήμα του περιεχομένου του ΚΕ που αποτελεί τον κωδικό λειτουργίας της εντολής οδηγείται στην μονάδα ελέγχου για να αποκωδικοποιηθεί. Από την περιγραφή των εντολών, που έχουμε κάνει, είναι σαφές ότι οι εντολές ADD, SUB και STORE απαιτούν την ανάγνωση του περιεχομένου των καταχωρητών, οι αριθμοί αναφοράς των οποίων δηλώνονται στα πεδία r1 και r2 των εντολών. Επίσης η εντολή LOAD απαιτεί την ανάγνωση του περιεχομένου του καταχωρητή ο αριθμός αναφοράς του οποίου δηλώνεται στο πεδίο r2. Αφενός η θέση και το εύρος των πεδίων r1 και r2 είναι η ίδια και στις τέσσερις εντολές, αφ' ετέρου η ανάγνωση ενός καταχωρητή από μόνη της δεν δημιουργεί κανένα πρόβλημα, διότι μπορούμε απλά να αγνοήσουμε τα δεδομένα που εμφανίζονται στην έξοδό του. Επομένως στο δεύτερο κύκλο μπορεί να γίνει ταυτόχρονα με την αποκωδικοποίηση της εντολής, ανάγνωση των καταχωρητών που δηλώνονται στα πεδία r1 και r2. Θεωρούμε ότι οι καταχωρητές γενικού σκοπού είναι κατασκευασμένες κατά τέτοιο τρόπο ώστε στις πόρτες εξόδου τους ΔΑ₁ και ΔΑ₂ να εμφανίζονται πάντοτε τα δεδομένα που περιέχονται στους καταχωρητές των οποίων οι διευθύνσεις οδηγούν τις εισόδους διευθύνσεων ανάγνωσης δΑ₁ και δΑ₂ των καταχωρητών. Άρα δεν απαιτείται κάποιο σήμα ελέγχου για την ενεργοποίηση της ανάγνωσης των καταχωρητών. Στο σχήμα 3.12.3 δίνεται η ροή της πληροφορίας και οι τιμές των σημάτων ελέγχου κατά την διάρκεια του 2^{ου} κύκλου ρολογιού του κύκλου εντολής.

Οι ενέργειες που θα λάβουν χώρα κατά την διάρκεια των επόμενων κύκλων ρολογιού, επομένως και οι τιμές των σημάτων ελέγχου που θα παραχθούν από την μονάδα ελέγχου σ' αυτούς τους κύκλους ρολογιού, καθορίζονται από την εντολή που έχει προσκομιστεί στον πρώτο κύκλο ρολογιού και αποκωδικοποιηθεί στο δεύτερο κύκλο ρολογιού. Θα εξετάσουμε λοιπόν τους υπόλοιπους κύκλους ρολογιού ανάλογα με το είδος της εντολής που εκτελείται.

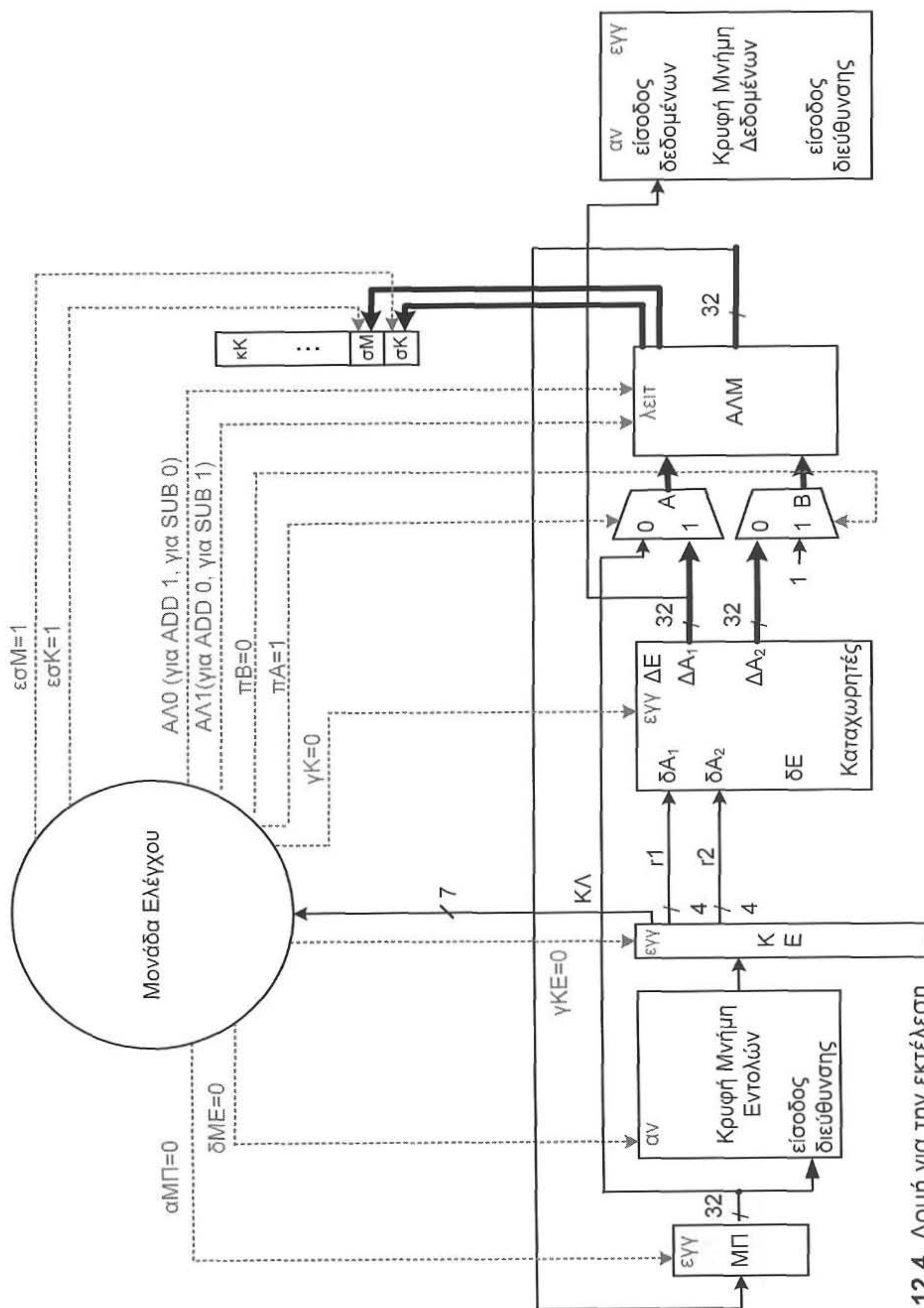


Σχήμα 3.12.3. Δομή για την εκτέλεση του 2ου κύκλου ρολογιού.

Κατ' αρχήν θα εξετάσουμε την περίπτωση η εκτελούμενη εντολή να είναι μια από τις εντολές ADD και SUB.

3^{ος} κύκλος ρολογιού (εντολή ADD η SUB).

Για να εκτελεστεί η πράξη της πρόσθεσης ή της αφαίρεσης οι έξοδοι δεδομένων ΔA_1 και ΔA_2 των πορτών ανάγνωσης δεδομένων πρέπει να οδηγήσουν τις εισόδους της ΑΛΜ. Λαμβάνοντας υπόψη ότι οι εισόδοι της ΑΛΜ οδηγούνται ήδη από την έξοδο του ΜΠ και την αριθμητική τιμή 1, θα πρέπει να τοποθετήσουμε δύο πολυπλέκτες Α και Β στις εισόδους της ΑΛΜ. Τα σήματα ελέγχου πA και πB των πολυπλεκτών Α και Β αντίστοιχα θα έχουν τις τιμές 1 και 0 (βλέπε σχήμα 3.12.4), οπότε τα δεδομένα, τα οποία είναι διαθέσιμα από τον προηγούμενο κύκλο στις εξόδους δεδομένων ΔA_1 και ΔA_2 των καταχωρητών, οδηγούν τις εισόδους της ΑΛΜ. Η αριθμητική λογική μονάδα δέχεται επίσης τα σήματα ελέγχου ΑΛ0 και ΑΛ1, οι τιμές των οποίων καθορίζουν την πράξη που θα εκτελέσει. Στην περίπτωση που η εκτελούμενη εντολή είναι η πρόσθεση η μονάδα ελέγχου θέτει τα σήματα ελέγχου ΑΛ1 και ΑΛ0 στις τιμές 0 και 1 αντίστοιχα. Όταν η εκτελούμενη εντολή είναι η αφαίρεση η μονάδα ελέγχου θέτει τα σήματα ελέγχου ΑΛ1 και ΑΛ0 στις τιμές 1 και 0 αντίστοιχα. Ο χρόνος καθυστέρησης της ΑΛΜ είναι μικρότερος της διάρκειας της περιόδου του σήματος χρονισμού, οπότε στο τέλος του 3^{ου} κύκλου ρολογιού τα αποτελέσματα είναι διαθέσιμα στην έξοδο της ΑΛΜ. Όταν εκτελείται μία αριθμητική πράξη κατά την διάρκεια του 3^{ου} κύκλου ρολογιού τα σήματα ελέγχου εσΜ και εσΚ έχουν την τιμή 1. Οπότε κατά την μετάβαση του σήματος χρονισμού από 0 σε 1 ενημερώνονται και οι σημαίες σΜ και σΚ του καταχωρητή κατάστασης κΚ ανάλογα με το αποτέλεσμα της πράξης που εκτελέστηκε. Στο σχήμα 3.12.4 δίνονται οι τιμές των σημάτων ελέγχου και η ροή της πληροφορίας κατά τη διάρκεια του 3^{ου} κύκλου ρολογιού του κύκλου εντολής.



Σχήμα 3.12.4. Δομή για την εκτέλεση του 3ου κύκλου εντολής ADD ή SUB

4^{ος} κύκλος ρολογιού (εντολή *ADD* ή *SUB*).

Στο τέλος του προηγούμενου κύκλου, δηλαδή του τρίτου κύκλου, το αποτέλεσμα της πράξης ήταν διαθέσιμο στην έξοδο της ΑΛΜ. Το αποτέλεσμα αυτό διατηρείται στην έξοδο της ΑΛΜ και κατά τη διάρκεια του 4^{ου} κύκλου ρολογιού διότι τις εισόδους της ΑΛΜ τροφοδοτούν τα ίδια δεδομένα, αρκεί τα σήματα ελέγχου που λαμβάνει η ΑΛΜ από την μονάδα ελέγχου να είναι τα ίδια. Κατά τη διάρκεια του 4^{ου} κύκλου ρολογιού το αποτέλεσμα της πράξης που εκτελέστηκε στην ΑΛΜ πρέπει να οδηγηθεί στην είσοδο δεδομένων ΔΕ της πόρτας εγγραφής των καταχωρητών γενικού σκοπού. Επίσης το περιεχόμενο του πεδίου r3 της εντολής που υπάρχει στον καταχωρητή εντολών ΚΕ, το οποίο δηλώνει τον καταχωρητή στον οποίο πρέπει να αποθηκευτεί το αποτέλεσμα, πρέπει να οδηγήσει την είσοδο διευθύνσεων ΔΕ της πόρτας εγγραφής των καταχωρητών γενικού σκοπού. Το σήμα ελέγχου γΚ, που οδηγεί την είσοδο ενεργοποίησης εγγραφής των καταχωρητών, πρέπει κατά τη διάρκεια του 4^{ου} κύκλου ρολογιού να έχει την τιμή 1. Επομένως κατά την μετάβαση του σήματος χρονισμού από 0 σε 1, το αποτέλεσμα γράφεται στον καταχωρητή γενικού σκοπού ο αριθμός του οποίου υπάρχει στο πεδίο r3 της εντολής. Στο σχήμα 3.12.5 δίνονται οι τιμές των σημάτων ελέγχου και η ροή της πληροφορίας κατά τη διάρκεια του 4^{ου} κύκλου ρολογιού του κύκλου εντολής.

Στη συνέχεια θα εξετάσουμε την περίπτωση που η εκτελούμενη εντολή είναι η εντολή *STORE*.

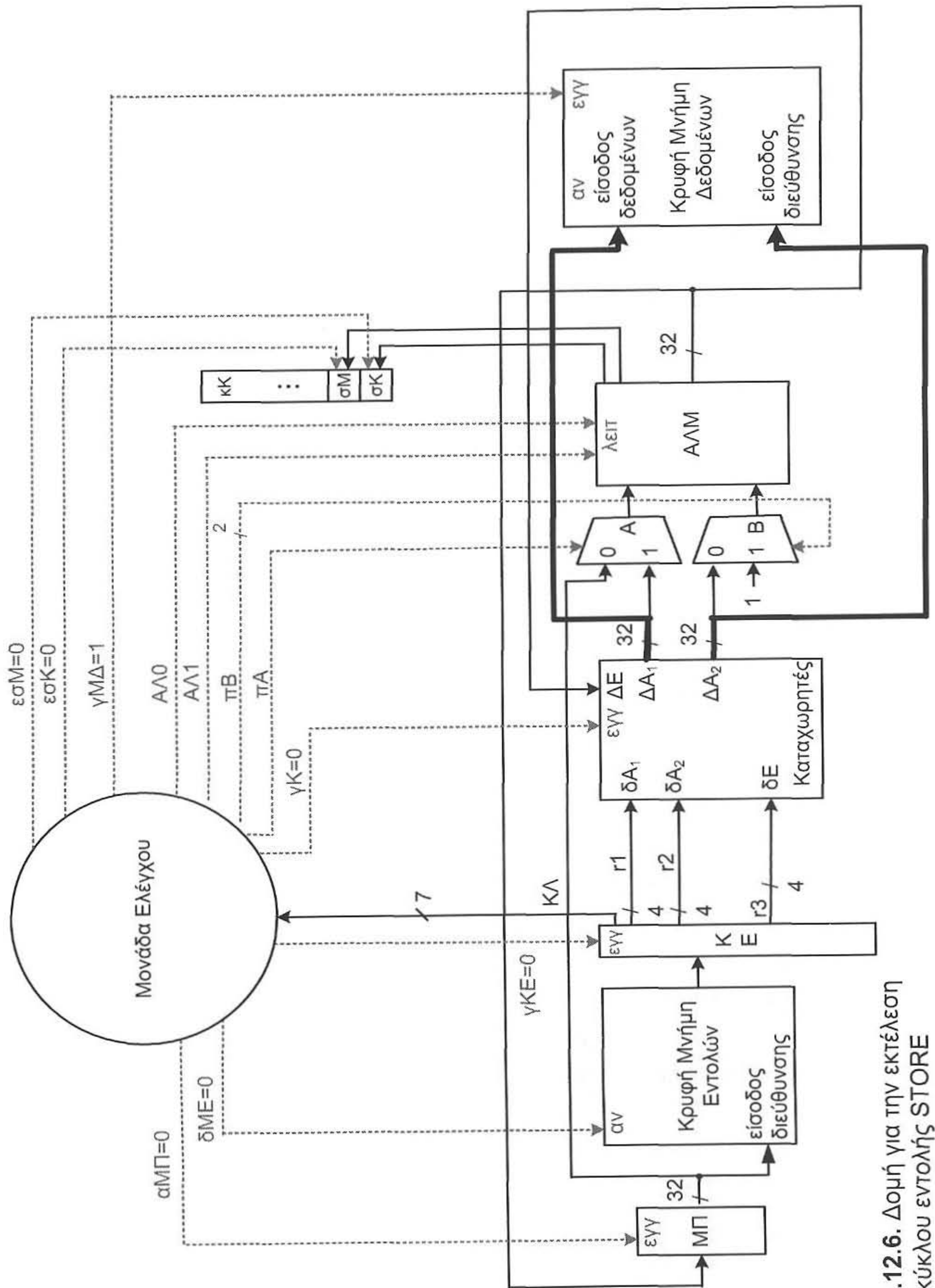
3^{ος} κύκλος ρολογιού (εντολή *STORE*).

Τα δεδομένα που είναι αποθηκευμένα στους καταχωρητές γενικού σκοπού, οι οποίοι δηλώνονται στα πεδία r1 και r2 της εντολής, είναι διαθέσιμα στις εξόδους δεδομένων ΔΑ₁ και ΔΑ₂ από το τέλος του δεύτερου κύκλου ρολογιού και πρέπει να οδηγηθούν στην είσοδο δεδομένων και την είσοδο διεύθυνσης αντίστοιχα της κρυφής μνήμης δεδομένων. Η είσοδος ενεργοποίησης εγγραφής εγγ της Κρυφής Μνήμης Δεδομένων οδηγείται από ένα σήμα γΜΔ το οποίο κατά τη διάρκεια του 3^{ου} κύκλου ρολογιού πρέπει να έχει την τιμή 1. Επομένως μετά από χρόνο ίσο με το χρόνο προσπέλασης της κρυφής μνήμης δεδομένων, τα δεδομένα που οδηγούν την είσοδο δεδομένων της κρυφής μνήμης δεδομένων αποθηκεύονται στη θέση της κρυφής μνήμης δεδομένων με διεύθυνση τη διεύθυνση που οδηγεί την είσοδο διεύθυνσής της. Έχει προβλεφτεί ο χρόνος εγγραφής στην κρυφή μνήμη δεδομένων να είναι μικρότερος της περιόδου του ρολογιού, οπότε στο τέλος του τρίτου κύκλου ρολογιού η εντολή *STORE* έχει ολοκληρωθεί. Στο σχήμα 3.12.6 δίνονται οι τιμές των σημάτων ελέγχου και η ροή της πληροφορίας κατά τη διάρκεια του 3^{ου} κύκλου ρολογιού του κύκλου εντολής.

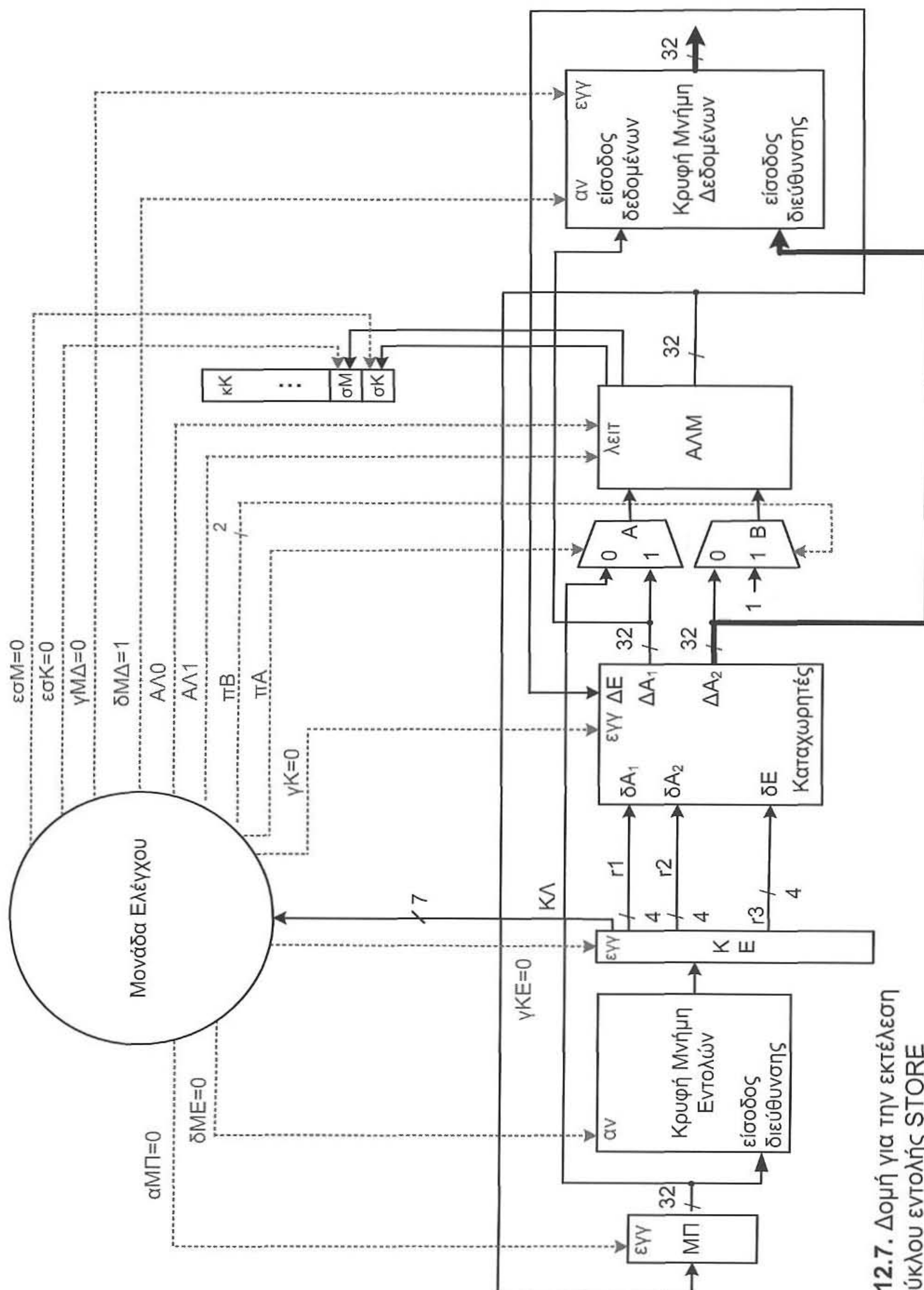
Στη συνέχεια θα εξετάσουμε την περίπτωση που η εκτελούμενη εντολή είναι η εντολή *LOAD*.

3^{ος} κύκλος ρολογιού (εντολή *LOAD*).

Τα δεδομένα που είναι διαθέσιμα στην ΔΑ₂ έξοδο της πόρτας ανάγνωσης δεδομένων των καταχωρητών γενικού σκοπού πρέπει να οδηγήσουν την είσοδο διεύθυνσης της Κρυφής Μνήμης Δεδομένων. Κατά την διάρκεια αυτού του κύκλου ρολογιού το σήμα ελέγχου δΜΔ πρέπει να έχει την τιμή 1, οπότε μετά από χρόνο ίσο με το χρόνο προσπέλασης της Κρυφής Μνήμης Δεδομένων, το περιεχόμενο της θέσης μνήμης με διεύθυνση αυτή που οδηγεί την είσοδο διεύθυνσης θα εμφανιστεί στις εξόδους της Κρυφής Μνήμης Δεδομένων. Για να δουλεύει σωστά η μονάδα μας πρέπει ο χρόνος προσπέλασης της Κρυφής Μνήμης Δεδομένων να είναι μικρότερος του χρόνου κύκλου ρολογιού. Στο σχήμα 3.12.7 δίνονται οι τιμές των σημάτων ελέγχου και η ροή της πληροφορίας κατά τη διάρκεια του 3^{ου} κύκλου ρολογιού του κύκλου της εντολής *LOAD*.



Σχήμα 3.12.6. Δομή για την εκτέλεση του 3ου κύκλου εντολής STORE

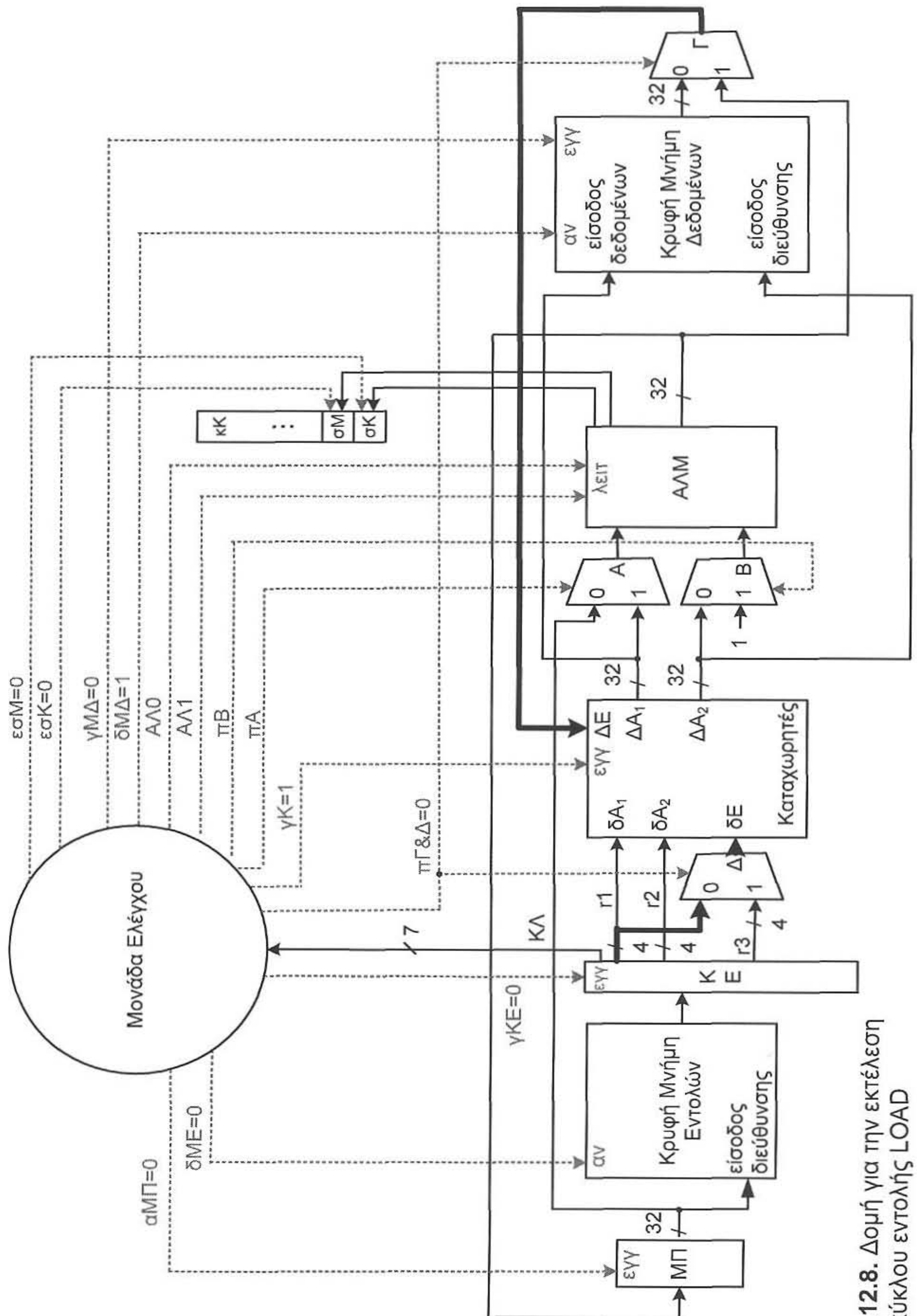


Σχήμα 3.12.7. Δομή για την εκτέλεση του 3ου κύκλου εντολής STORE

4^{ος} κύκλος ρολογιού (εντολή *LOAD*).

Κατά τη διάρκεια του 4^{ου} κύκλου ρολογιού, τα δεδομένα που οδηγούν την είσοδο διεύθυνσης της κρυφής μνήμης δεδομένων δεν αλλάζουν και η τιμή του σήματος ελέγχου δΜΔ συνεχίζει να έχει την τιμή 1, επομένως στην έξοδο της κρυφής μνήμης διατηρείται το περιεχόμενο της θέσης που προσπελάστηκε στον προηγούμενο κύκλο ρολογιού. Στον 4^ο κύκλο ρολογιού το πεδίο r1 της εντολής που εκτελείται πρέπει να οδηγήσει την είσοδο διεύθυνσης δΕ της πόρτας εγγραφής των καταχωρητών, ενώ την είσοδο δεδομένων ΔΕ της πόρτας εγγραφής πρέπει να οδηγεί η έξοδος δεδομένων της κρυφής μνήμης δεδομένων. Λαμβάνοντας υπόψη ότι η είσοδος δεδομένων ΔΕ της πόρτας εγγραφής των καταχωρητών γενικού σκοπού οδηγείται ήδη από την έξοδο της ΑΛΜ, θα προσθέσουμε ένα πολυπλέκτη στην είσοδο ΔΕ, τον πολυπλέκτη Γ. Κατά τον 4^ο κύκλο ρολογιού το σήμα ελέγχου του πολυπλέκτη Γ πρέπει να έχει τη λογική τιμή 0 (βλέπε σχήμα 3.12.8). Επίσης η είσοδος διευθύνσεων δΕ της πόρτας εγγραφής των καταχωρητών γενικού σκοπού οδηγείται από το πεδίο r3 της εντολής που εκτελείται, επομένως πρέπει και στην είσοδο δΕ να προσθέσουμε ένα πολυπλέκτη Δ. Οι πολυπλέκτες Γ και Δ μπορούν να έχουν το ίδιο σήμα ελέγχου πΓ&Δ, το οποίο κατά την διάρκεια του κύκλου αυτού θα έχει την λογική τιμή 0. Το σήμα ελέγχου γΚ, που οδηγεί την είσοδο ενεργοποίησης εγγραφής των καταχωρητών γενικού σκοπού πρέπει να έχει την τιμή 1, ώστε στο τέλος του τρέχοντος κύκλου ρολογιού κατά την μετάβαση του σήματος χρονισμού από 0 σε 1 να αποθηκευτεί στον καταχωρητή, που δηλώνεται στο πεδίο r1 της εντολής, το περιεχόμενο της θέσης της κρυφής μνήμης δεδομένων, η διεύθυνση της οποίας υπάρχει στον καταχωρητή που δηλώνεται στο πεδίο r2 της εντολής. Στο σχήμα 3.12.8 δίνονται οι τιμές των σημάτων ελέγχου και η ροή της πληροφορίας κατά τη διάρκεια του 4^{ου} κύκλου ρολογιού του κύκλου εντολής.

Στη συνέχεια θα εξετάσουμε την περίπτωση η εκτελούμενη εντολή να είναι μια από τις εντολές BRZ και BRC.



Σχήμα 3.12.8. Δομή για την εκτέλεση του 4ου κύκλου εντολής LOAD

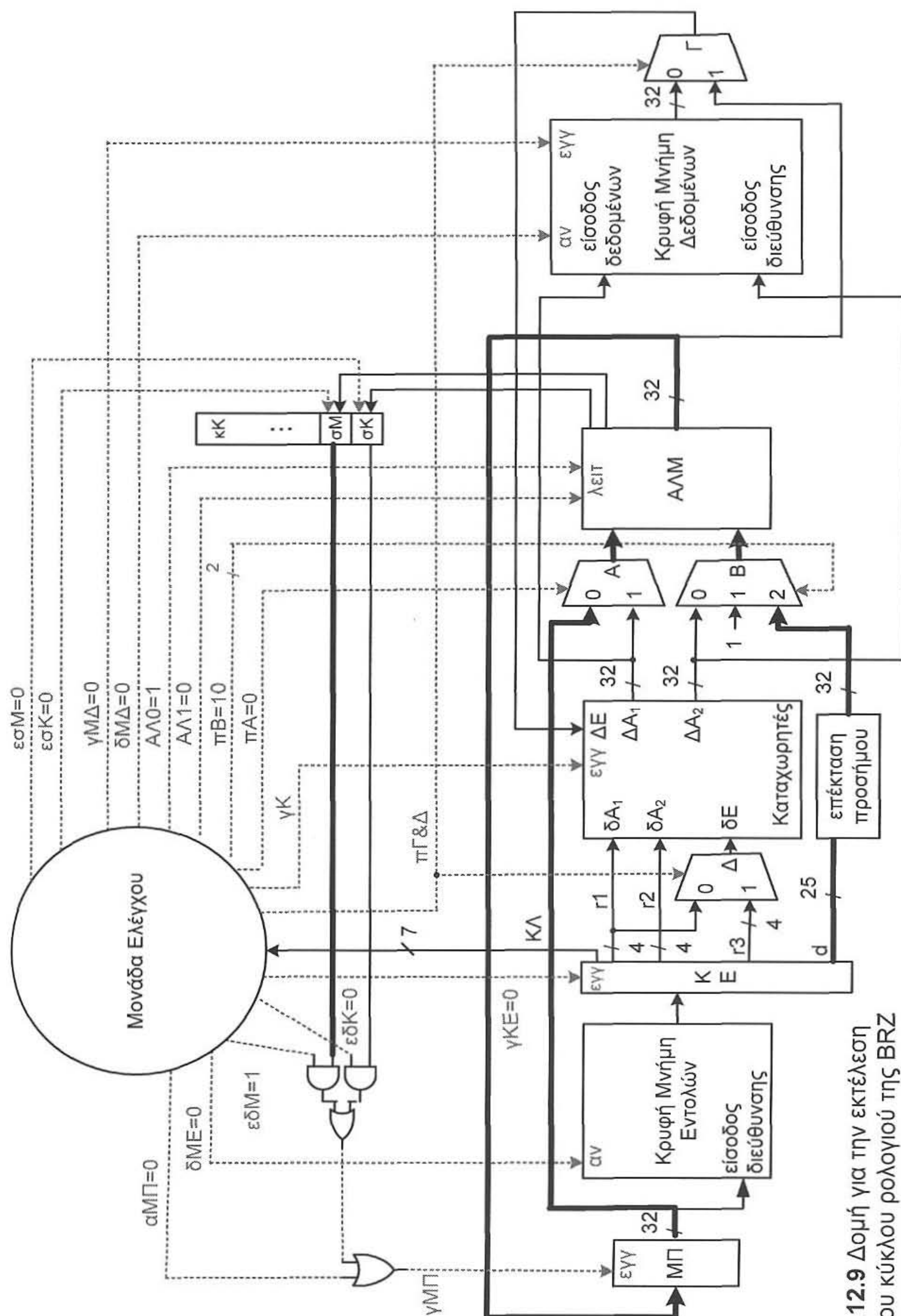
3^{ος} κύκλος ρολογιού (εντολή BRZ ή BRC).

Τα περιεχόμενα των καταχωρητών των οποίων οι αριθμοί αναφοράς δηλώνονται στα πεδία r1 και r2 της εντολής που εκτελείται είναι διαθέσιμα στις εξόδους δεδομένων ΔΑ₁ και ΔΑ₂ του αρχείου καταχωρητών από το τέλος του προηγούμενου κύκλου ρολογιού, ωστόσο δεν χρειάζονται σ' αυτή την εντολή και δεν θα χρησιμοποιηθούν. Τα 25 λιγότερο σημαντικά δυαδικά ψηφία του ΚΕ, τα οποία αποτελούν τον αριθμό μετατόπισης της εντολής, πρέπει να προστεθούν με το περιεχόμενο του ΜΠ. Για να γίνει η πρόσθεση θα πρέπει να γίνει επέκταση προσήμου του αριθμού μετατόπισης σε 32 δυαδικά ψηφία. Επίσης επειδή πρέπει να οδηγήσουμε την είσοδο της ΑΛΜ με τον αριθμό μετατόπισης, πρέπει να αντικαταστήσουμε τον πολυπλέκτη Β με ένα πολύπλεκτη 3 σε 1, οπότε ο πολυπλέκτης Β θα έχει δύο εισόδους ελέγχου που θα συμβολίζουμε πάλι με το πΒ. Κατά τον τρέχοντα κύκλο ρολογιού τα σήματα ελέγχου πΒ του πολυπλέκτη Β έχουν τις τιμές 10, οπότε ο αριθμός μετατόπισης οδηγεί την μία είσοδο της ΑΛΜ. Το σήμα ελέγχου πΑ του πολυπλέκτη Α έχει την τιμή 0, επομένως η άλλη είσοδος της ΑΛΜ τροφοδοτείται με το τρέχον περιεχόμενο του ΜΠ (βλέπε σχήμα 3.12.9). Μετά από χρόνο ίσο με το χρόνο καθυστέρησης της ΑΛΜ στις εξόδους της θα υπάρχει το άθροισμα ΜΠ+d. Η τιμή των σημάτων ελέγχου εσΜ και εσΚ πρέπει να είναι 0, ώστε ανεξάρτητα του αποτελέσματος της ΑΛΜ, να μην ενημερώνονται οι σημαίες σΜ και σΚ του καταχωρητή κατάστασης κΚ. Το εάν θα αποθηκευτεί η τιμή του ΜΠ+d στον ΜΠ ή όχι εξαρτάται από το εάν ικανοποιείται συνθήκη της εντολής διακλάδωσης που εκτελείται, δηλαδή εάν η αντίστοιχη σημαία του καταχωρητή κατάστασης έχει την τιμή 1.

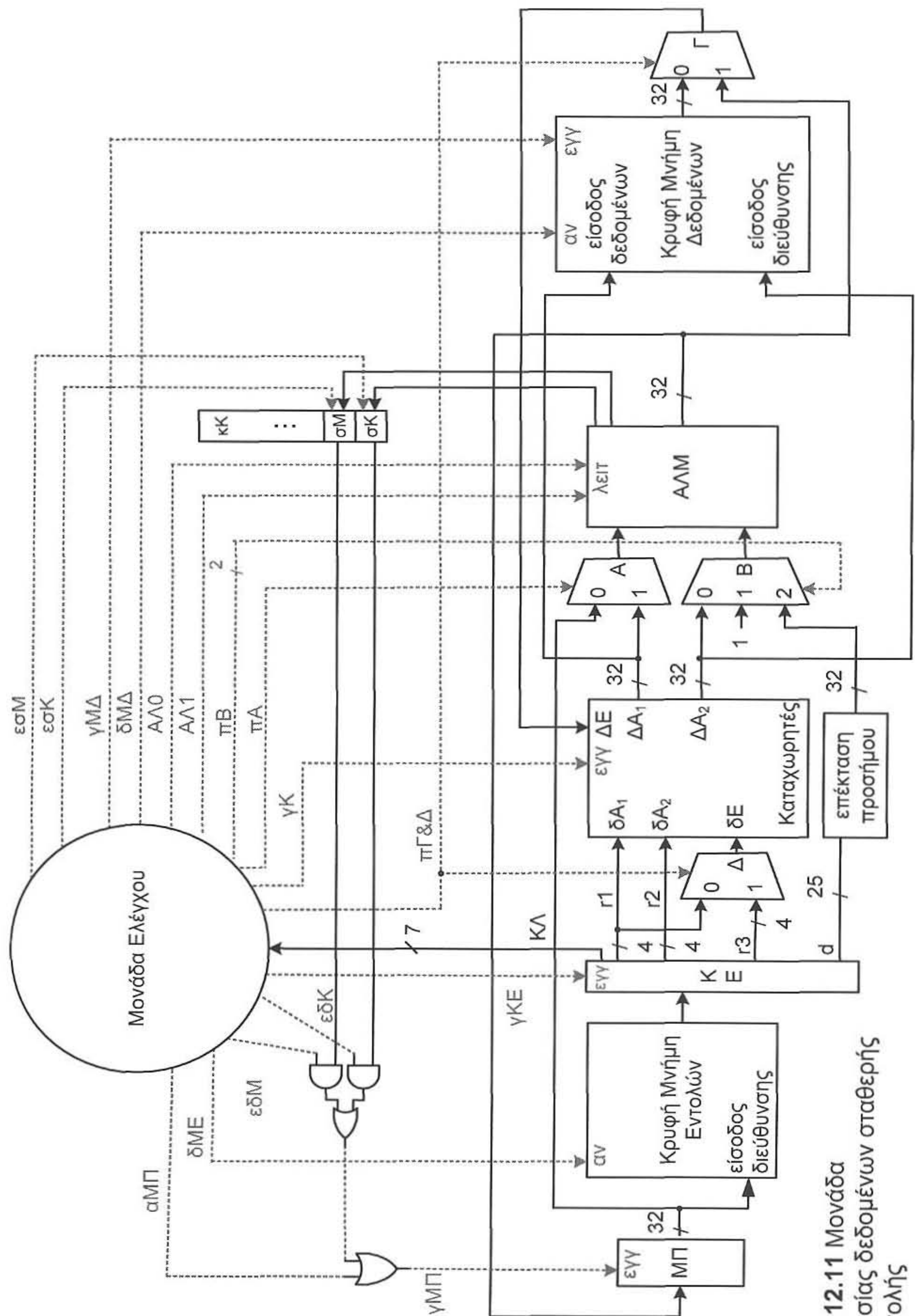
Εάν η εκτελούμενη εντολή είναι η BRZ και η τιμή της σημαίας σΜ έχει την τιμή 1 ή η εκτελούμενη εντολή είναι η BRC και η τιμή της σημαίας σΚ έχει την τιμή 1, τότε κατά τη μετάβαση του σήματος χρονισμού από 0 σε 1 στον ΜΠ πρέπει να αποθηκευτεί η τιμή ΜΠ+d που υπάρχει στις εξόδους της ΑΛΜ. Επομένως σ' αυτή την περίπτωση η είσοδος εγγραφής εγγ του ΜΠ θα πρέπει κατά την μετάβαση του σήματος χρονισμού να έχει την λογική τιμή 1. Για το σκοπό αυτό η μονάδα ελέγχου κατά την διάρκεια του τρέχοντος κύκλου θέτει, ανάλογα με την εκτελούμενη εντολή το σήμα ελέγχου εδΜ ή το σήμα ελέγχου εδΚ στην τιμή 1.

Στα σχήματα 3.12.9 και 3.12.10 δίνονται οι τιμές των σημάτων ελέγχου και η ροή της πληροφορίας κατά τη διάρκεια του 3^{ου} κύκλου ρολογιού όταν η εκτελούμενη εντολή είναι η εντολή BRZ και BRC αντίστοιχα.

Στο σχήμα 3.12.11 δίνεται η δομή της μονάδας επεξεργασίας δεδομένων που μας ζητήθηκε.



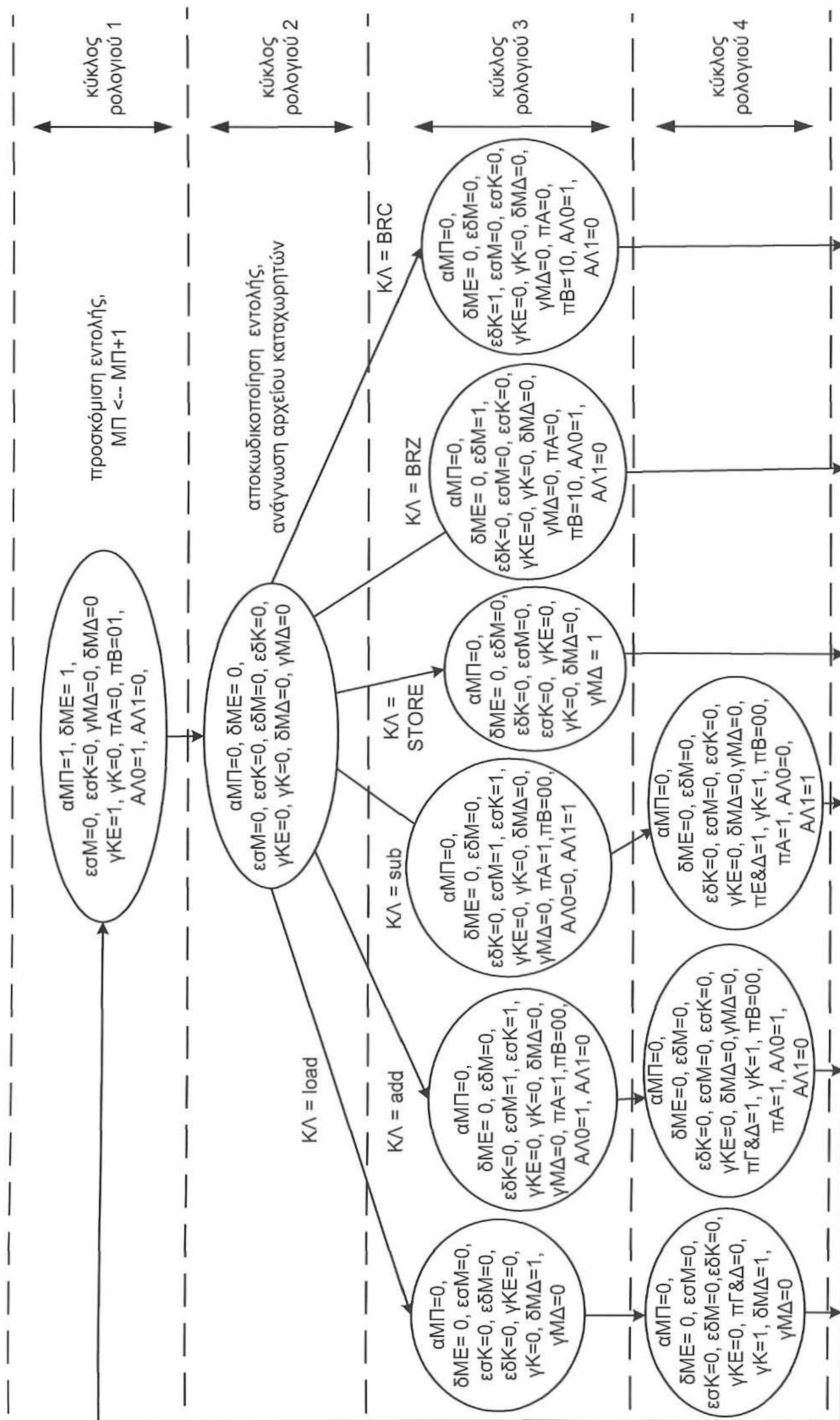
Σχήμα 3.12.9 Δομή για την εκτέλεση και του 3ου κύκλου ρολογιού της BRZ



Σχήμα 3.12.11 Μονάδα επεξεργασίας δεδομένων σταθερής υποδιαστολής

Όπως έχουμε εξηγήσει στο βιβλίο που αναφέραμε στην αρχή της άσκησης, τα σήματα ελέγχου ενός πολυπλέκτη ή αποπολυπλέκτη μπορούν να πάρουν οποιαδήποτε τιμή όταν ο συγκεκριμένος πολυπλέκτης ή αποπολυπλέκτης δεν χρησιμοποιείται κατά την διάρκεια ενός κύκλου ρολογιού. Ομοίως τα σήματα επιλογής συγκεκριμένης λειτουργίας μίας λειτουργικής μονάδας ή ενεργοποίησης ανάγνωσης μνήμης μπορούν να πάρουν οποιαδήποτε τιμή, κατά την διάρκεια ενός κύκλου ρολογιού που αυτή η λειτουργική μονάδα δεν χρησιμοποιείται. Ωστόσο εάν θέλουμε να διατηρούνται τα αποτελέσματα στις εξόδους της λειτουργικής μονάδας ή της μνήμης και στον επόμενο κύκλο ρολογιού τότε αφενός θα πρέπει και στο επόμενο κύκλο ρολογιού τα σήματα ελέγχου της υπομονάδας (μνήμη ή ΑΛΜ κ.λπ.) να έχουν τις ίδιες τιμές, αφετέρου οι είσοδοί της πρέπει να οδηγούνται με τα ίδια δεδομένα. Αντίθετα, τα σήματα ελέγχου ενεργοποίησης εγγραφής σε καταχωρητή ή μνήμη πρέπει να έχουν την ενεργό τιμή (εμείς έχουμε θεωρήσει ως ενεργό τιμή την τιμή 1) σ' εκείνους τους κύκλους στο τέλος των οποίων θέλουμε να γίνει εγγραφή και σ' όλους τους άλλους κύκλους να έχουν την τιμή 0.

Στο σχήμα 3.12.12 δίνεται το διάγραμμα καταστάσεων της Μονάδας Ελέγχου. Από το διάγραμμα καταστάσεων μπορούμε να σχεδιάσουμε την Μονάδα Ελέγχου ως κλασικό ακολουθιακό κύκλωμα ή με την τεχνική του μικροπρογραμματισμού.



Σχήμα 3.12.12 Διάγραμμα καταστάσεων της Μονάδας Ελέγχου.

$t_{εκ}(M)$: ελάχιστη καθυστέρηση της μονάδας M (contamination delay). Δηλώνει τον ελάχιστο χρόνο που περνάει από τη στιγμή που στις εισόδους της μονάδας έχουμε εφαρμόσει νέες τιμές μέχρι τη στιγμή που τουλάχιστον μια έξοδος της μονάδας έχει αλλάξει τιμή.

$t_{\pi}(M)$: χρόνος προ-τοποθέτησης (set-up time).

$t_{\delta}(M)$: χρόνος διατήρησης (hold time).

Στη συνέχεια δίνονται οι χρόνοι που σχετίζονται με κάθε υπομονάδα της μονάδας επεξεργασίας.

Μετρητής προγράμματος

$t_{\mu\kappa}(ΜΠ)$: 80 ps, $t_{εκ}(ΜΠ)$: 30 ps, $t_{\pi}(ΜΠ)$: 50 ps και $t_{\delta}(ΜΠ)$: 0 ps

Κρυφή μνήμη εντολών (Κρυφ.ΜΕ)

$t_{\mu\kappa}(Κρυφ.ΜΕ)$: 6 ns, $t_{εκ}(Κρυφ.ΜΕ)$: 5 ns

Κρυφή μνήμη δεδομένων (Κρυφ.ΜΔ)

$t_{\mu\kappa}(Κρυφ.ΜΔ)$: 6 ns, $t_{εκ}(Κρυφ.ΜΔ)$: 5 ns

Πολυπλέκτης 2 σε 1, (Π2-1)

$t_{\mu\kappa}(Π2-1)$: 80 ps, $t_{εκ}(Π2-1)$: 40 ps (θεωρούμε ότι οι χρόνοι είναι οι ίδιοι για αλλαγή μιας εισόδου δεδομένων ή της εισόδου ελέγχου)

Πολυπλέκτης 3 σε 1, (Π3-1)

$t_{\mu\kappa}(Π3-1)$: 90 ps, $t_{εκ}(Π3-1)$: 50 ps (θεωρούμε ότι οι χρόνοι είναι οι ίδιοι για αλλαγή μιας εισόδου δεδομένων ή εισόδου ελέγχου).

Μονάδα ελέγχου, (ΜΕλ.)

$t_{\mu\kappa}(ΜΕλ.)$: 90 ps, $t_{εκ}(ΜΕλ.)$: 60 ps

Καταχωρητές γενικού σκοπού

$t_{\mu\kappa}(Καταχ.)$: 1 ns, $t_{εκ}(Καταχ.)$: 300 ps, $t_{\pi}(Καταχ.)$: 50 ps και $t_{\delta}(Καταχ.)$: 0 ps

Αριθμητική-Λογική μονάδα

$t_{\mu\kappa}(ΑΛΜ)$: 4 ns, $t_{εκ}(ΑΛΜ)$: 30 ps, $t_{\mu\kappa}(αποτέλεσμα\ 0)$: 4040 ps

Μονάδα επέκτασης προσήμου (Επ. Π)

$t_{\mu\kappa}(Επ.Π)$: 4 ps, $t_{εκ}(Επ.Π)$: 0 ps

Πύλη AND

$t_{\mu\kappa}(AND)$: 40 ps, $t_{εκ}(AND)$: 0 ps

Πύλη OR

$t_{\mu\kappa}(OR)$: 40 ps, $t_{εκ}(OR)$: 0 ps

Όλοι οι καταχωρητές που χρησιμοποιούνται (εκτός των καταχωρητών γενικού σκοπού) έχουν τους ίδιους χρόνους με τον Μετρητή Προγράμματος.

Να υπολογίσετε την ελάχιστη περίοδο του σήματος χρονισμού (ρολόι).

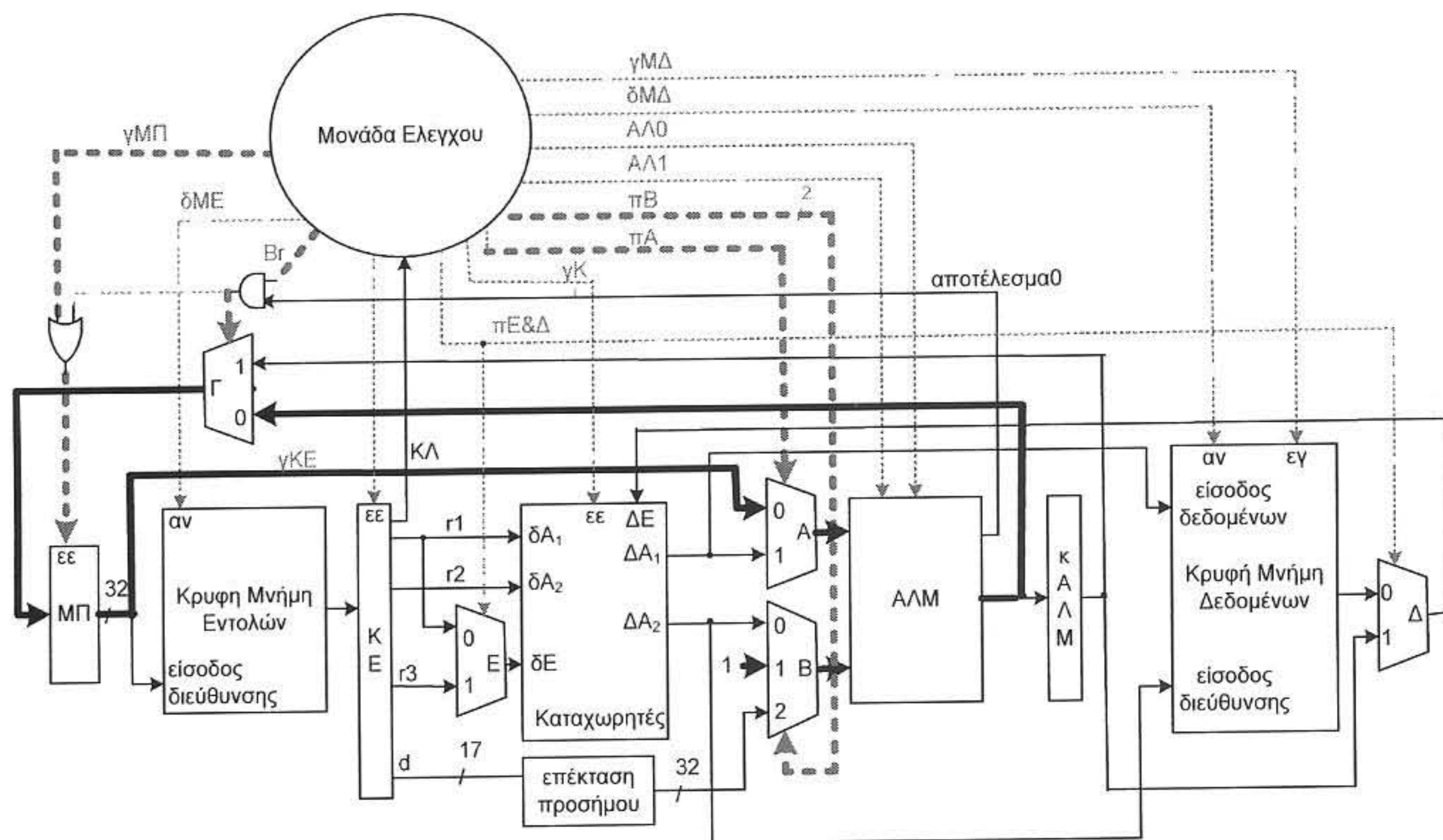
Απάντηση

Θα υπολογίσουμε το χρόνο που απαιτείται για να εκτελεστούν όλες οι λειτουργίες που πρέπει να εκτελεστούν σε κάθε κύκλο ρολογιού κατά την εκτέλεση κάθε μίας από τις εντολές που μελετάμε και θα πάρουμε ως περίοδο του κύκλου ρολογιού τον μέγιστο χρόνο που θα υπολογίσουμε. Θα χρησιμοποιήσουμε το συμβολισμό $T_{\text{εντολή}}^i$ για να δηλώνουμε το χρόνο που απαιτείται για την ολοκλήρωση των λειτουργιών του i κύκλου της εντολής «εντολή». Όπως έχουμε δει στην υποενότητα 3.3.2 του βιβλίου οι λειτουργίες που λαμβάνουν χώρα στους δύο πρώτους κύκλους κατά την προσκόμιση και εκτέλεση μιας εντολής δεν εξαρτώνται από την εντολή.

Για να υπολογίσουμε τον ελάχιστο χρόνο που απαιτείται για να ολοκληρωθούν όλες οι ενέργειες που πρέπει να λάβουν χώρα σε ένα κύκλο ρολογιού κατά την εκτέλεση μιας εντολής, θα πρέπει να βρούμε την ενέργεια που απαιτεί το μεγαλύτερο χρονικό διάστημα σ' αυτό τον κύκλο. Για την εκτέλεση μιας ενέργειας μπορεί να απαιτείται η ροή περισσότερων του ενός σημάτων ελέγχου και περισσότερων της μιας μεταφορών δεδομένων. Επομένως, για να βρούμε την ενέργεια η οποία απαιτεί το μεγαλύτερο χρονικό διάστημα, πρέπει για κάθε ενέργεια να βρούμε το μονοπάτι διάδοσης σήματος ελέγχου ή και δεδομένων που βάζει τη μεγαλύτερη καθυστέρηση. Για το σκοπό αυτό θα υπολογίζουμε την καθυστέρηση όλων των δυνατών μονοπατιών που συμμετέχουν στην εν λόγω ενέργεια. Η εύρεση όλων των δυνατών μονοπατιών που συμμετέχουν σε μία συγκεκριμένη ενέργεια, γίνεται εύκολα εάν ξεκινάμε από το τέλος ενός μονοπατιού και ακολουθούμε όλες τις δυνατές διαδρομές έως την πηγή του σήματος ελέγχου ή δεδομένων. Για να διευκολυνθούμε στην ανεύρεση όλων των δυνατών μονοπατιών, σε κάθε ένα από τα σχήματα που θα ακολουθήσουν δίνεται με φαρδύτερες γραμμές η ροή των δεδομένων (συνεχής γραμμή) και των σημάτων ελέγχου (διακεκομμένη γραμμή) που συμμετέχουν στη διαδικασία που περιγράφεται στο συγκεκριμένο σχήμα.

Κατά τη διάρκεια του πρώτου κύκλου ρολογιού, κατά την εκτέλεση μιας εντολής, λαμβάνουν χώρα δύο ανεξάρτητες ενέργειες:

1. Η ανάγνωση της κρυφής μνήμης εντολών και η αποθήκευση της εντολής που διαβάστηκε στον καταχωρητή εντολών, ΚΕ.
2. Η αύξηση του περιεχομένου του ΜΠ κατά ένα.



Σχήμα 3.13.3 Πρώτος κύκλος ρολογιού κατά την εκτέλεση εντολής: $MP=MP+1$.

Στο σχήμα 3.13.3 δίνονται με φαρδύτερες γραμμές τα σήματα που συμμετέχουν στη διαδικασία αύξησης του περιεχομένου του ΜΠ κατά ένα.

$$T^1(MP=MP+1) =$$

$$= \max \{ [t_{\pi}(MP) + t_{\mu\kappa}(OR) + t_{\mu\kappa}(ME\lambda.)], \\ [t_{\pi}(MP) + t_{\mu\kappa}(\Pi\Gamma') + t_{\mu\kappa}(AND) + t_{\mu\kappa}(ME\lambda.)], \\ [t_{\pi}(MP) + t_{\mu\kappa}(\Pi\Gamma') + t_{\mu\kappa}(A\Lambda M) + t_{\mu\kappa}(\Pi A) + t_{\mu\kappa}(ME\lambda.)], \\ [t_{\pi}(MP) + t_{\mu\kappa}(\Pi\Gamma') + t_{\mu\kappa}(A\Lambda M) + t_{\mu\kappa}(\Pi A) + t_{\mu\kappa}(MP)], \\ [t_{\pi}(MP) + t_{\mu\kappa}(\Pi\Gamma') + t_{\mu\kappa}(A\Lambda M) + t_{\mu\kappa}(\Pi B) + t_{\mu\kappa}(ME\lambda.)] \} =$$

$$= \max \{ [50 \text{ ps} + 40 \text{ ps} + 90 \text{ ps}], \\ [50 \text{ ps} + 80 \text{ ps} + 40 \text{ ps} + 90 \text{ ps}], \\ [50 \text{ ps} + 80 \text{ ps} + 4 \text{ ns} + 80 \text{ ps} + 90 \text{ ps}], \\ [50 \text{ ps} + 80 \text{ ps} + 4 \text{ ns} + 80 \text{ ps} + 80 \text{ ps}], \\ [50 \text{ ps} + 80 \text{ ps} + 4 \text{ ns} + 90 \text{ ps} + 90 \text{ ps}] \} =$$

$$= \max \{ [180 \text{ ps}], [260 \text{ ps}], [4.300 \text{ ps}], [4.290 \text{ ps}], [4.310 \text{ ps}] \} = 4.310 \text{ ps}$$

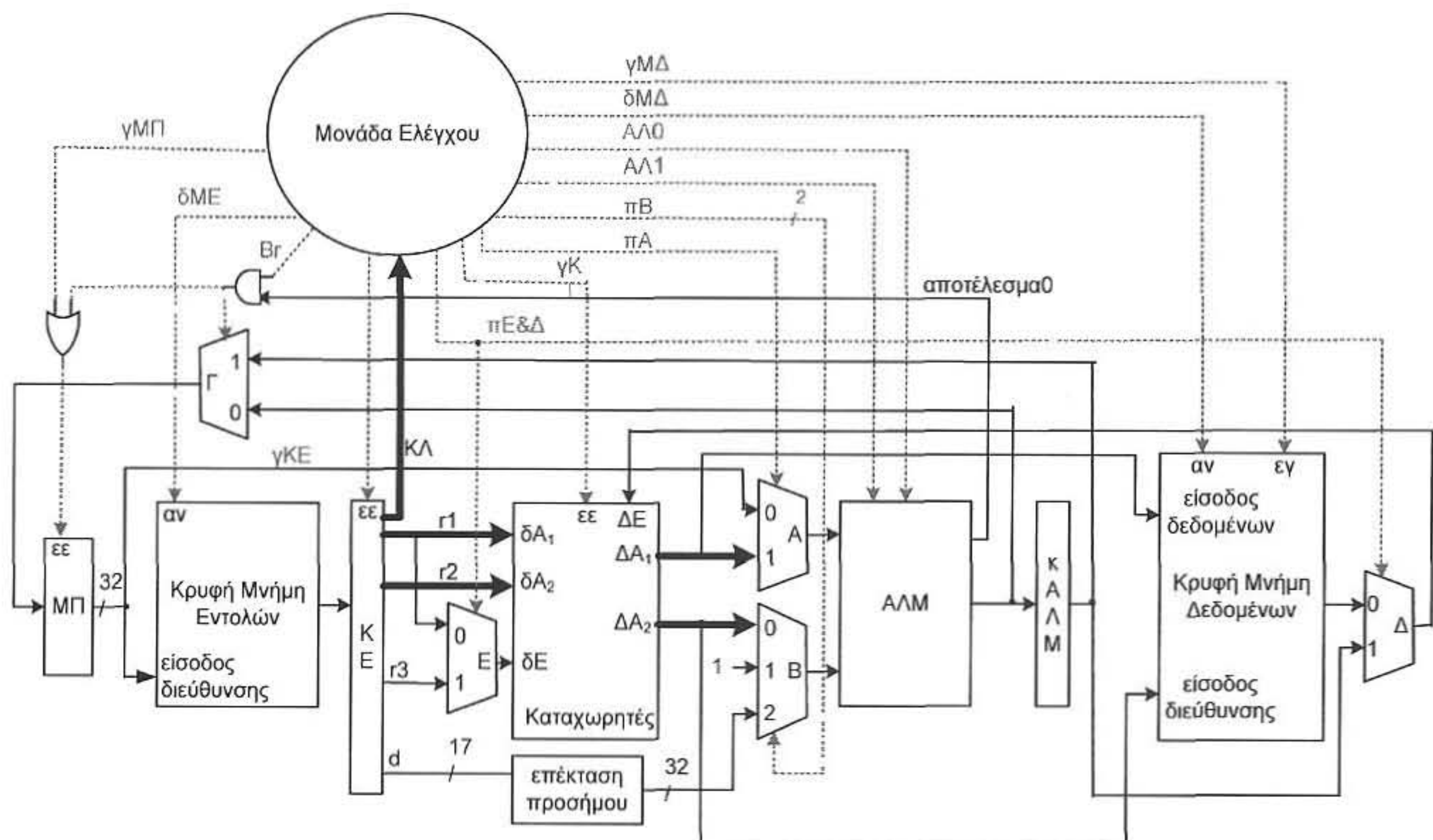
Επομένως $T^1 =$

$$= \max \{ T^1(\text{ανάγνωση κρυφής μνήμης εντολών και αποθήκευση στον ΚΕ}), \\ T^1(MP=MP+1) \} = \max \{ 6.140 \text{ ps}, 4.310 \text{ ps} \} = 6.140 \text{ ps}.$$

Κατά τη διάρκεια του δεύτερου κύκλου ρολογιού, κατά την εκτέλεση μιας εντολής, λαμβάνουν χώρα δύο ανεξάρτητες ενέργειες:

1. Ανάγνωση των καταχωρητών γενικού σκοπού.
2. Υπολογισμός του ΜΠ+d

Στο σχήμα 3.13.4 δίνονται με φαρδύτερες γραμμές τα σήματα που συμμετέχουν στη διαδικασία ανάγνωσης των καταχωρητών γενικού σκοπού.

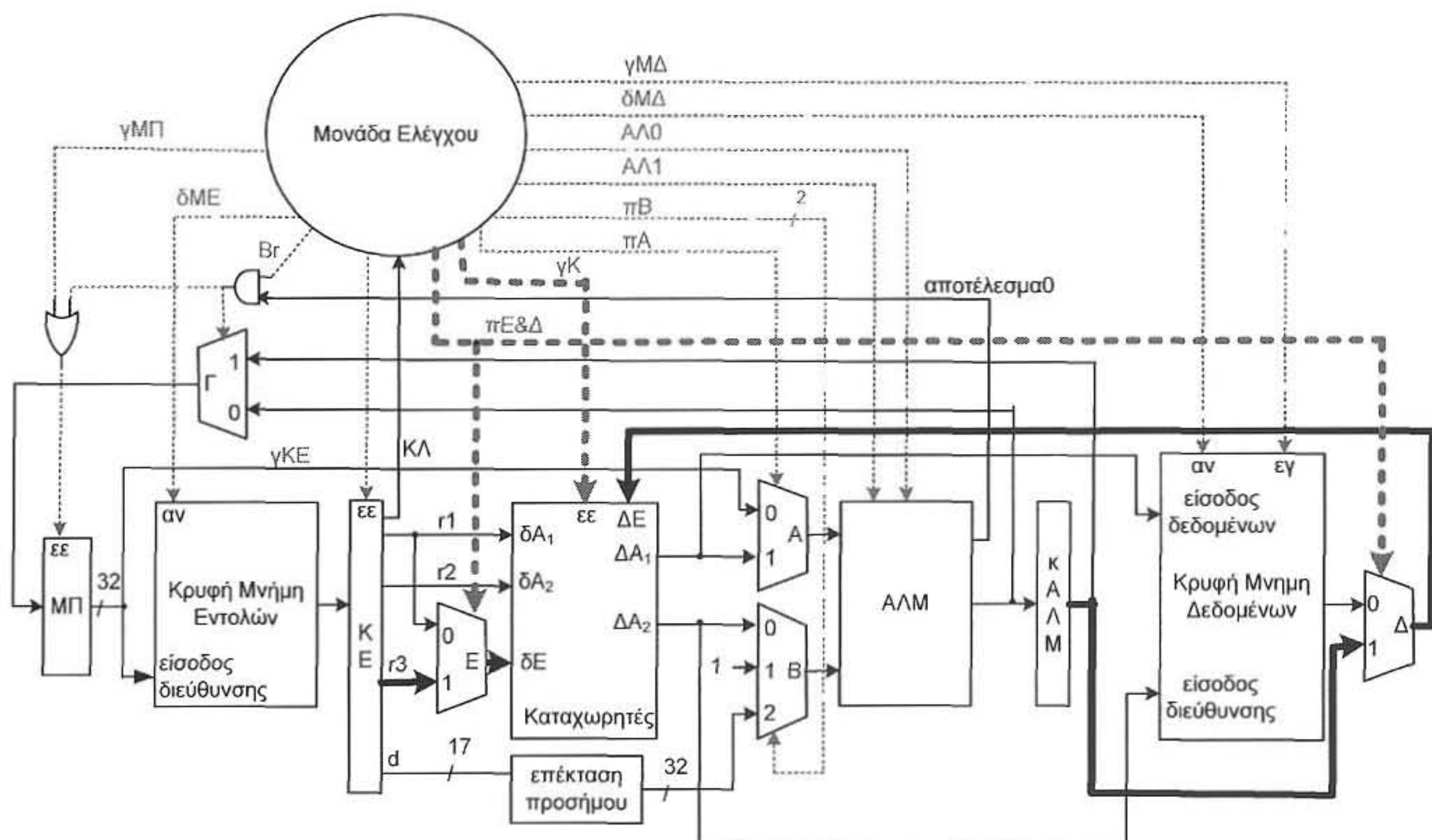


Σχήμα 3.13.4 Δεύτερος κύκλος ρολογιού κατά την εκτέλεση εντολής: Ανάγνωση κατά χωρητών

$$T^2 (\text{ανάγνωση των καταχωρητών γενικού σκοπού}) = t_{\mu\kappa}(\text{Καταχ.}) + t_{\mu\kappa}(\text{ΚΕ}) = 1 \text{ ns} + 80 \text{ ps} = 1080 \text{ ps}$$

Στο σχήμα 3.13.5 δίνονται με φαρδύτερες γραμμές τα σήματα που συμμετέχουν στη διαδικασία υπολογισμού του ΜΠ+d.

$$\begin{aligned}
 &= \max \{ [50 \text{ ps} + 80 \text{ ps}], \\
 &\quad [50 \text{ ps} + 80 \text{ ps} + 90 \text{ ps}], \\
 &\quad [50 \text{ ps} + 90 \text{ ps}], \\
 &\quad [50 \text{ ps} + 80 \text{ ps} + 90 \text{ ps}], \\
 &\quad [50 \text{ ps} + 80 \text{ ps} + 80 \text{ ps}] \} = \\
 &= \max \{ [130 \text{ ps}], [220 \text{ ps}], [140 \text{ ps}], [220 \text{ ps}], [210 \text{ ps}] \} = 220 \text{ ps}
 \end{aligned}$$



Σχήμα 3.13.11 Τέταρτος κύκλος ρολογιού κατά την εκτέλεση της εντολής ADD, SUB ή AND.

Στον επόμενο Πίνακα 3.13.1 δίνεται η ελάχιστη διάρκεια που πρέπει να έχει κάθε κύκλος ανάλογα με την εντολή που εκτελείται για να προλαβαίνουν να εκτελεστούν όλες οι απαιτούμενες λειτουργίες. Η μέγιστη τιμή που εμφανίζεται στον Πίνακα 3.13.1 καθορίζει την ελάχιστη περίοδο του σήματος χρονισμού (ρολόι). Επομένως η περίοδος του σήματος χρονισμού πρέπει να είναι μεγαλύτερη ή ίση του $T=6140 \text{ ps}$.

Πίνακας 3.13.1

κύκλος	LOAD	STORE	ADD, SUB ή AND	BRE
T ₁	6.140 ps	6.140 ps	6.140 ps	6.140 ps
T ₂	4.230 ps	4.230 ps	4.230 ps	4.230 ps
T ₃	6.090 ps	6090 ps	4.230 ps	4.350 ps
T ₄	220 ps	-	220 ps	-

ΚΕΦΑΛΑΙΟ

4

Ασκήσεις για επεξεργαστές μερικώς επικαλυπτόμενων λειτουργιών

Άσκηση 4.1

Θεωρήστε ένα μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών των N βαθμίδων. Στο μηχανισμό αυτό εκτελείται μια διαδικασία για την εκτέλεση της οποίας σε σύστημα που δεν υλοποιεί την τεχνική των μερικώς επικαλυπτόμενων λειτουργιών απαιτούνται K ns. Θεωρήστε ότι δεν εμφανίζονται εξαρτήσεις και ότι η διαδικασία μπορεί να διαιρεθεί σε N βήματα ίσης διάρκειας, κάθε ένα των οποίων εκτελείται σε μια βαθμίδα του μηχανισμού.

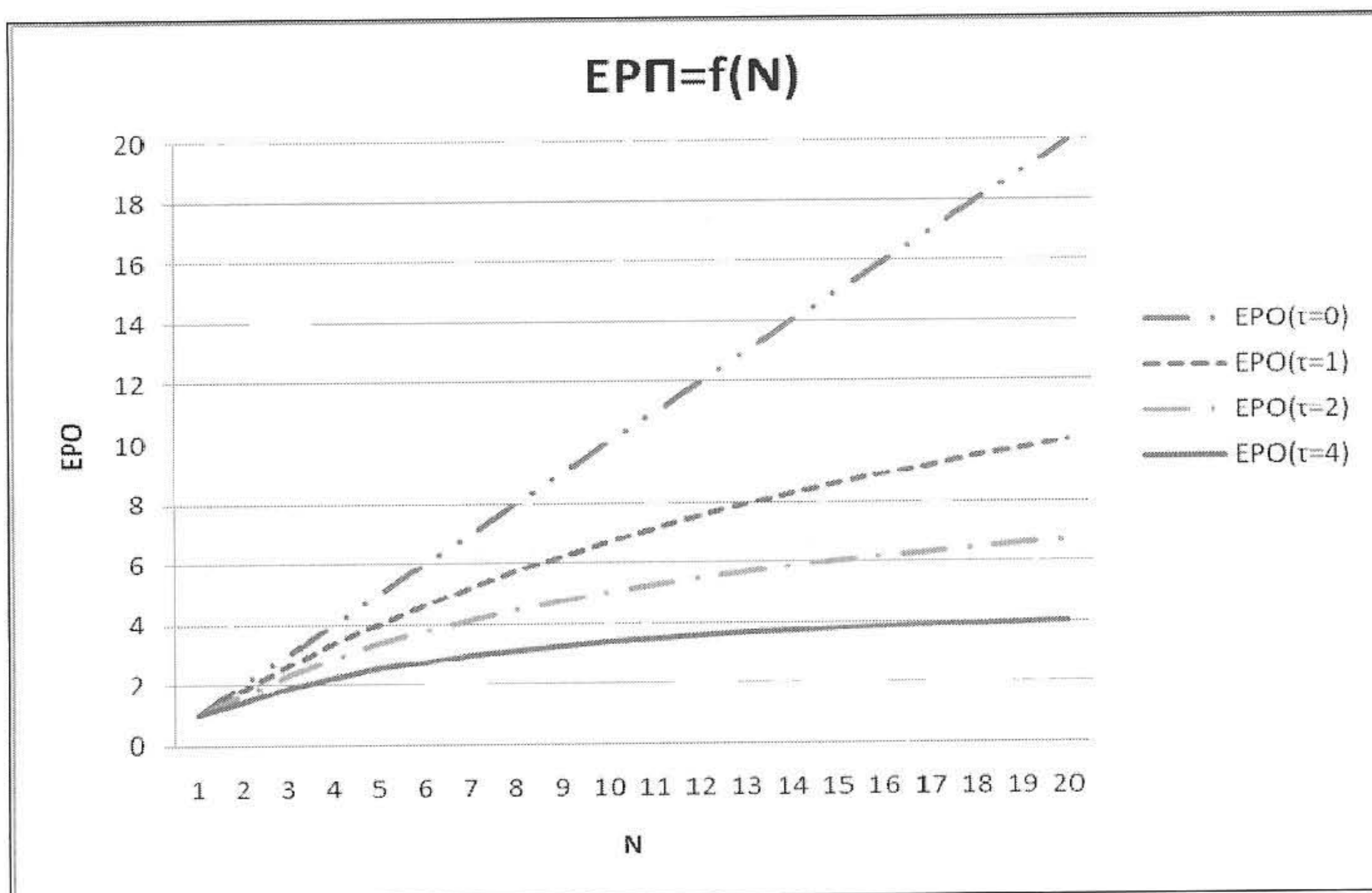
Για $K=20$ ns και $K=40$ ns να δώσετε σχεδιάγραμμα της επιτάχυνσης του ρυθμού ολοκλήρωσης διαδικασιών EPO, την οποία επιτυγχάνει ο εν λόγω μηχανισμός, ως συνάρτηση του πλήθους των βαθμίδων N , με παράμετρο την καθυστέρηση τ των καταχωρητών του μηχανισμού για $\tau = 0$ ns, 1 ns, 2 ns και 4 ns. Να σχολιάσετε τα σχήματα. Αγνοήστε τις αρχικές συνθήκες.

Απάντηση

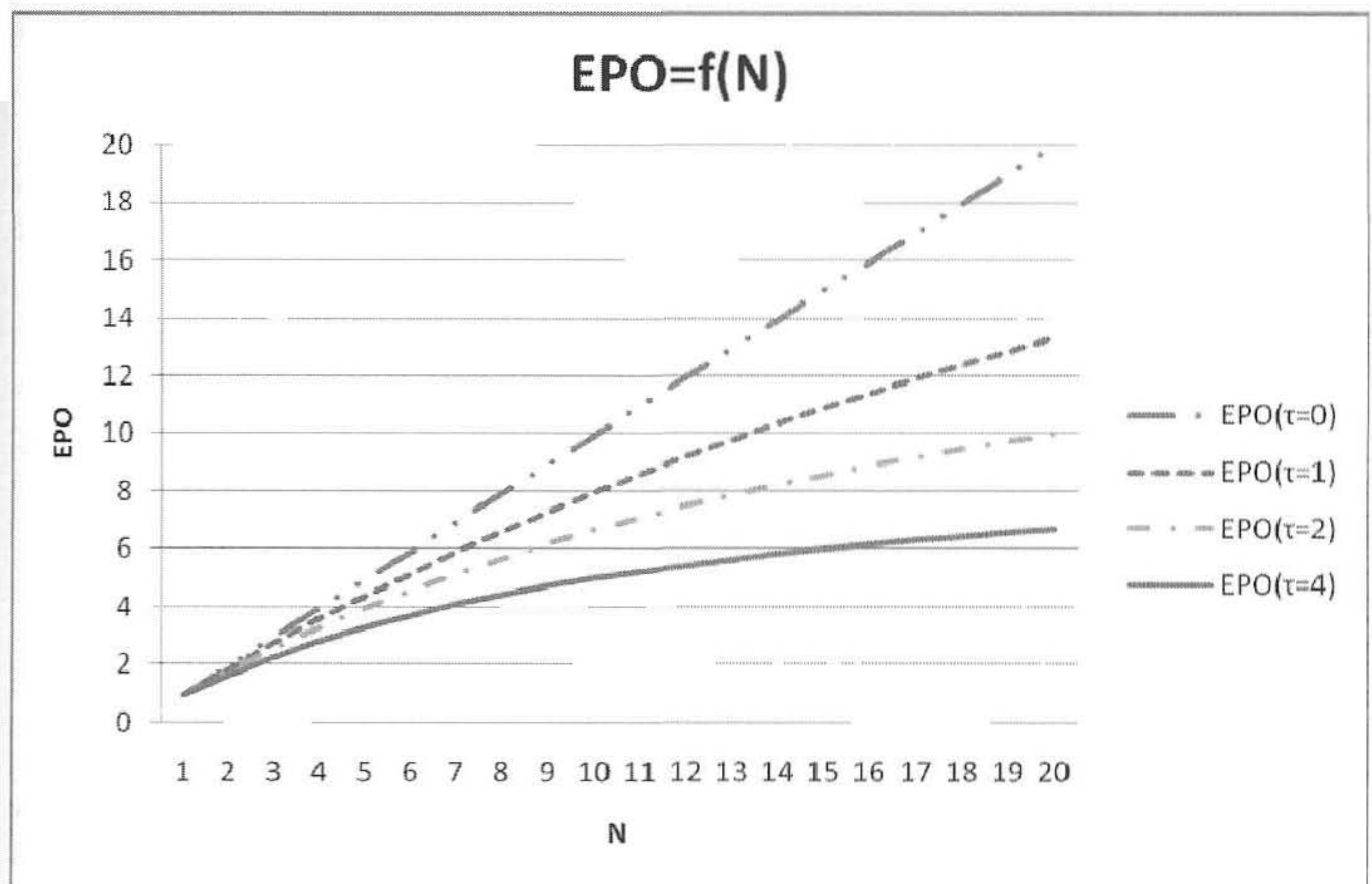
Λαμβάνοντας υπ' όψη ότι η διαδικασία μπορεί να διαιρεθεί σε N βήματα ίσης διάρκειας, καθένα εκ των οποίων εκτελείται σε μια βαθμίδα του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών, συμπεραίνουμε ότι στο μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών των N βαθμίδων, μη λαμβάνοντας υπόψη τις αρχικές συνθήκες, θα ολοκληρώνεται μια διαδικασία κάθε $K/N + \tau$ ns. Επομένως η επιτάχυνση του ρυθμού ολοκλήρωσης διαδικασιών EPO θα ισούται με $EPO = K/(K/N + \tau) = (K \times N)/(K + N \times \tau)$. Για $\tau=0$ έχουμε τη μέγιστη τιμή $EPO=N$. Στα σχήματα 4.1.1 και 4.1.2 δίνονται οι τιμές της EPO για $K=20$ και $K=40$ αντίστοιχα.

Παρατηρούμε ότι όσο μεγαλύτερη είναι η καθυστέρηση των καταχωρητών του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών, τόσο μεγαλύτερη είναι η διαφορά της τιμής της EPO που επιτυγχάνεται, από τη μέγιστη τιμή N που έχουμε για $\tau = 0$. Από τα σχήματα 4.1.1 και 4.1.2 επίσης παρατηρούμε, ότι από ένα σημείο και μετά δεν έχει νόημα η αύξηση του πλήθους των βαθμίδων του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών διότι η αύξηση της τιμής της EPO είναι πάρα πολύ μικρή, ενώ όπως γνωρίζουμε οδηγεί σε αύξηση του χρόνου που απαιτείται για την εκτέλεση μιας διαδικασίας (latency).

Επίσης συγκρίνοντας τα σχήματα 4.1.1 και 4.1.2 παρατηρούμε ότι όσο μεγαλύτερη είναι η τιμή του λόγου τ/K , τόσο μεγαλύτερη είναι η απόκλιση της τιμής του EPO που επιτυγχάνεται από τη μέγιστη τιμή N . Επίσης παρατηρούμε ότι όσο μεγαλύτερη είναι η τιμή του λόγου τ/K , τόσο για μικρότερη τιμή του N φτάνει στον κόρο η τιμή του $EPO(N)$.



Σχήμα 4.1.1 Η επιτάχυνση του ρυθμού ολοκλήρωσης διαδικασιών EPO, ως συνάρτηση του πλήθους των βαθμίδων N του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών, με παράμετρο την καθυστέρηση τ των καταχωρητών του μηχανισμού για $\tau = 0$ ns, 1 ns, 2 ns και 4 ns. Ο χρόνος εκτέλεσης της διαδικασίας σε σύστημα που δεν υλοποιεί την τεχνική των μερικώς επικαλυπτόμενων λειτουργιών είναι $K=20$ ns.



Σχήμα 4.1.2 Η επιτάχυνση του ρυθμού ολοκλήρωσης διαδικασιών EPO, ως συνάρτηση του πλήθους των βαθμίδων N του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών, με παράμετρο την καθυστέρηση τ των καταχωρητών του μηχανισμού για $\tau = 0$ ns, 1 ns, 2 ns και 4 ns. Ο χρόνος εκτέλεσης της διαδικασίας σε σύστημα που δεν υλοποιεί την τεχνική των μερικώς επικαλυπτόμενων λειτουργιών είναι $K=40$ ns.

Άσκηση 4.2 (4.1 βιβλίου)

Για την εκτέλεση μιας διαδικασίας απαιτείται η εκτέλεση της ακολουθίας βημάτων α , β , γ , δ , ϵ , ζ , η και θ . Τα βήματα αυτά μπορούν να εκτελεστούν από αντίστοιχες μονάδες A, B, Γ, Δ, E, Z, H και Θ με καθυστέρηση, 10 ns, 5 ns, 30 ns, 10 ns, 10 ns, 5 ns, 8 ns και 9 ns αντίστοιχα.

α. Να σχεδιάσετε το μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών (pipelining) που επιτυγχάνει το μέγιστο ρυθμό ολοκλήρωσης διαδικασιών και γι' αυτό το ρυθμό, τον ελάχιστο χρόνο ολοκλήρωσης μιας διαδικασίας (latency). Να δικαιολογήσετε την σχεδίαση.

β. Να υπολογίσετε το ρυθμό ολοκλήρωσης διαδικασιών και το χρόνο που απαιτείται για την ολοκλήρωση μιας διαδικασίας.

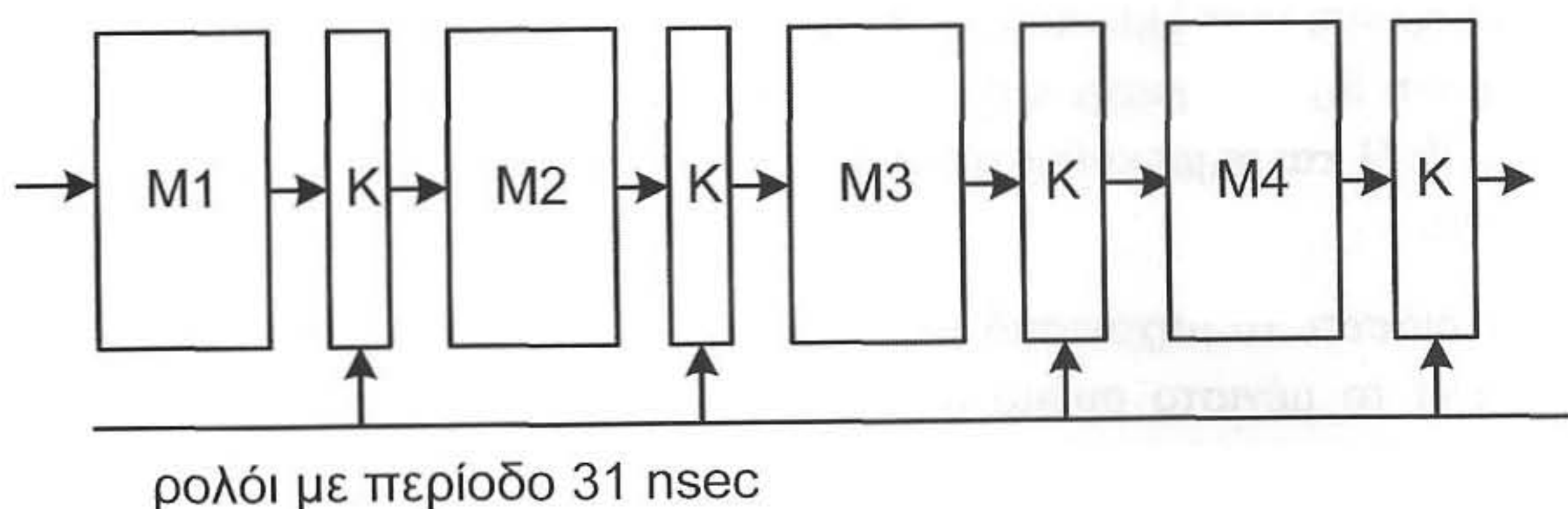
Έχετε στη διάθεσή σας καταχωρητές που η καθυστέρησή τους είναι 1 nsec.

Απάντηση

α.

Επειδή η άσκηση για κάθε βήμα της διαδικασίας μας δίνει και μία μονάδα, δεν επιτρέπεται να θεωρήσουμε ότι μπορούμε κάποιο βήμα της διαδικασίας να το διαιρέσουμε σε ένα μεγαλύτερο αριθμό από επιμέρους βήματα και να σχεδιάσουμε υπομονάδες για τα επιμέρους βήματα. Παρατηρούμε ότι η μονάδα Γ έχει τη μεγαλύτερη καθυστέρηση 30 nsec. Επομένως, αν θέλουμε να έχουμε το μέγιστο ρυθμό ολοκλήρωσης διαδικασιών στο μηχανισμό επικαλυπτόμενων λειτουργιών, θα πρέπει να αποτελέσει από μόνη της μία βαθμίδα. Αυτή η βαθμίδα θα καθορίσει τη συχνότητα του ρολογιού και το ρυθμό ολοκλήρωσης διαδικασιών. Λαμβάνοντας υπόψη ότι μεταξύ κάθε δύο γειτονικών βαθμίδων τοποθετείται ένας καταχωρητής και το γεγονός ότι ο καταχωρητής που έχουμε στη διάθεσή μας βάζει καθυστέρηση 1 nsec, συμπεραίνουμε ότι η περίοδος του κύκλου ρολογιού θα ισούται με $30+1 \text{ nsec} = 31 \text{ nsec}$.

Αφού θέλουμε χωρίς να μειώσουμε το ρυθμό ολοκλήρωσης διαδικασιών να πετύχουμε τον ελάχιστο χρόνο ολοκλήρωσης μιας διαδικασίας θα πρέπει το πλήθος των βαθμίδων που θα χρησιμοποιήσουμε να είναι το μικρότερο δυνατό. Αφού έχουμε 8 μονάδες θα πρέπει κάποιες από αυτές να μπουν στην ίδια βαθμίδα. Για να μη μειώσουμε το ρυθμό ολοκλήρωσης διαδικασιών, η συνολική καθυστέρηση κάθε βαθμίδας δεν θα πρέπει να ξεπερνάει τα 30 nsec (μη λαμβάνοντας υπόψη την καθυστέρηση του καταχωρητή). Επομένως οι μονάδες Α και Β θα αποτελέσουν την βαθμίδα M1, η μονάδα Γ από μόνη της θα αποτελέσει τη βαθμίδα M2, οι μονάδες Δ και Ε θα αποτελέσουν την βαθμίδα M3 και οι μονάδες Ζ, Η και Θ τη βαθμίδα M4. Εναλλακτικά οι μονάδες Δ, Ε και Ζ θα μπορούσαν να αποτελέσουν τη βαθμίδα M3 και οι μονάδες Η και Θ τη μονάδα M4. Ο μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών δίνεται στο σχήμα 4.2.1.



Σχήμα 4.2.1 Μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών.

β.

Ο ρυθμός ολοκλήρωσης διαδικασιών είναι μία διαδικασία κάθε 31 nsec.

Ο χρόνος που απαιτείται για την ολοκλήρωση μιας διαδικασίας ισούται με $31 \times 4 \text{ nsec} = 124 \text{ nsec}$.

Άσκηση 4.3 (4.2 βιβλίου)

Για την εκτέλεση μιας διαδικασίας N απαιτείται η εκτέλεση της ακολουθίας βημάτων $\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta$ και θ . Για την εκτέλεση των βημάτων διατίθενται οι αντίστοιχες μονάδες $A, B, \Gamma, \Delta, E, Z, H$ και Θ με καθυστέρηση, 19 nsec, 5 nsec, 10 nsec, 15 nsec, 9 nsec, 5, nsec, 8 nsec και 7 nsec αντίστοιχα.

α. Να σχεδιάσετε το μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών (pipelining) που επιτυγχάνει το μέγιστο ρυθμό ολοκλήρωσης διαδικασιών τύπου N , υπό την προϋπόθεση ότι ο χρόνος ολοκλήρωσης μιας διαδικασίας (latency) θα είναι μικρότερος των 105 nsec. Να δικαιολογήσετε τη σχεδίαση.

β. Να υπολογίσετε το ρυθμό ολοκλήρωσης διαδικασιών και το χρόνο που απαιτείται για την ολοκλήρωση μιας διαδικασίας.

Έχετε στη διάθεσή σας καταχωρητές που η καθυστέρησή τους είναι 1 nsec.

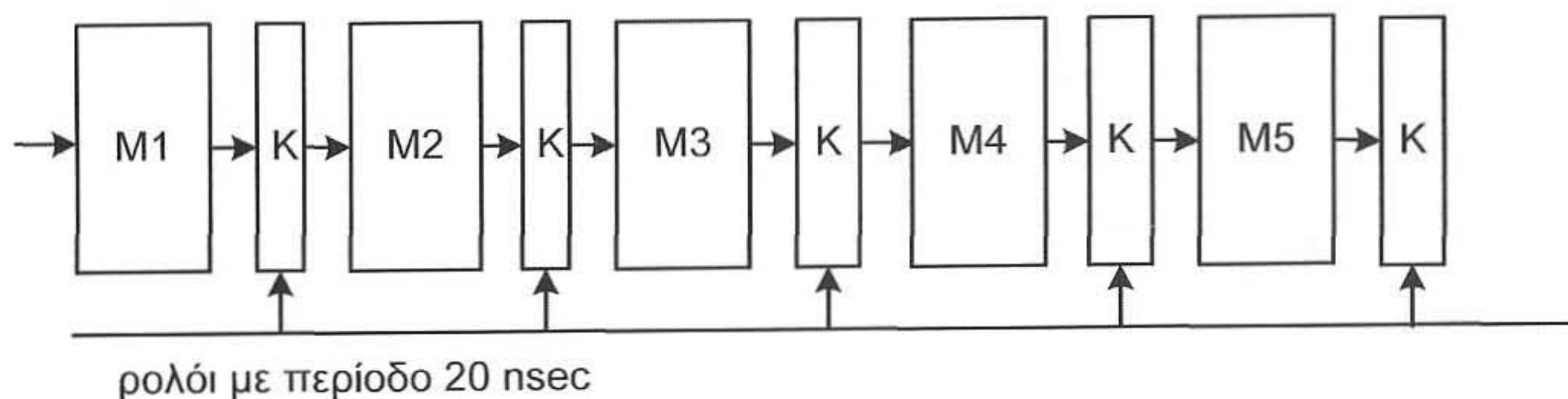
Απάντηση

α.

Παρατηρούμε ότι η μονάδα A έχει τη μεγαλύτερη καθυστέρηση 19 nsec. Επομένως, αν θέλουμε να έχουμε το μέγιστο ρυθμό ολοκλήρωσης διαδικασιών στο μηχανισμό επικαλυπτόμενων λειτουργιών, θα πρέπει να αποτελέσει από μόνη της μία βαθμίδα. Αυτή η βαθμίδα θα καθορίσει τη συχνότητα του ρολογιού και το ρυθμό ολοκλήρωσης διαδικασιών. Λαμβάνοντας υπόψη ότι μεταξύ κάθε δύο γειτονικών βαθμίδων τοποθετείται ένας καταχωρητής και το γεγονός ότι ο καταχωρητής που έχουμε στη διάθεσή μας βάζει καθυστέρηση 1 nsec, συμπεραίνουμε ότι η περίοδος του κύκλου ρολογιού θα ισούται με $19 + 1 \text{ nsec} = 20 \text{ nsec}$.

Επειδή θέλουμε ο χρόνος ολοκλήρωσης μιας διαδικασίας να είναι μικρότερος από 105 nsec θα πρέπει το πλήθος των βαθμίδων που θα χρησιμοποιήσουμε να είναι μικρότερο ή ίσο του $\lfloor 105/20 \rfloor = 5$. Αφού έχουμε 8 μονάδες θα πρέπει κάποιες από αυτές να μπουν στην ίδια βαθμίδα. Για να μη μειώσουμε το ρυθμό ολοκλήρωσης διαδικασιών η συνολική καθυστέρηση κάθε βαθμίδας δεν θα πρέπει να ξεπερνάει τα 19 nsec (μη λαμβάνοντας υπόψη την καθυστέρηση του καταχωρητή). Επομένως η μονάδα A , όπως

είπαμε, από μόνη της θα αποτελέσει τη βαθμίδα M1, οι μονάδες B και Γ θα αποτελέσουν την βαθμίδα M2, η μονάδα Δ θα αποτελέσει τη βαθμίδα M3, οι μονάδες E και Z τη βαθμίδα M4 και οι μονάδες H και Θ τη βαθμίδα M5. Ο μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών δίνεται στο σχήμα 4.3.1.



Σχήμα 4.3.1 Μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών.

β.

Ο ρυθμός ολοκλήρωσης διαδικασιών είναι μία διαδικασία κάθε 20 nsec.

Ο χρόνος που απαιτείται για την ολοκλήρωση μιας διαδικασίας ισούται με $20 \times 5 \text{ nsec} = 100 \text{ nsec}$.

Άσκηση 4.4

Για την εκτέλεση μιας διαδικασίας απαιτείται η εκτέλεση της ακολουθίας βημάτων α, β, γ, δ, ε, ζ, η, θ, κ και λ. Για την εκτέλεση των βημάτων διατίθενται οι αντίστοιχες μονάδες Α, Β, Γ, Δ, Ε, Ζ, Η, Θ, Κ και Λ με καθυστέρηση, 3 nsec, 4 nsec, 3 nsec, 5 nsec, 3 nsec, 4 nsec, 3 nsec, 3 nsec, 5 nsec και 4 nsec αντίστοιχα. Έχετε στη διάθεσή σας καταχωρητές που η καθυστέρησή τους είναι 1 nsec.

Να σχεδιάσετε το μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών (pipelining) που επιτυγχάνει το μέγιστο δυνατό ρυθμό ολοκλήρωσης διαδικασιών υπό την προϋπόθεση ότι η συνολική καθυστέρηση (latency) του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών είναι μικρότερη ή ίση των 44 nsec. Να δικαιολογήσετε τη σχεδίαση που θα προτείνετε.

Απάντηση

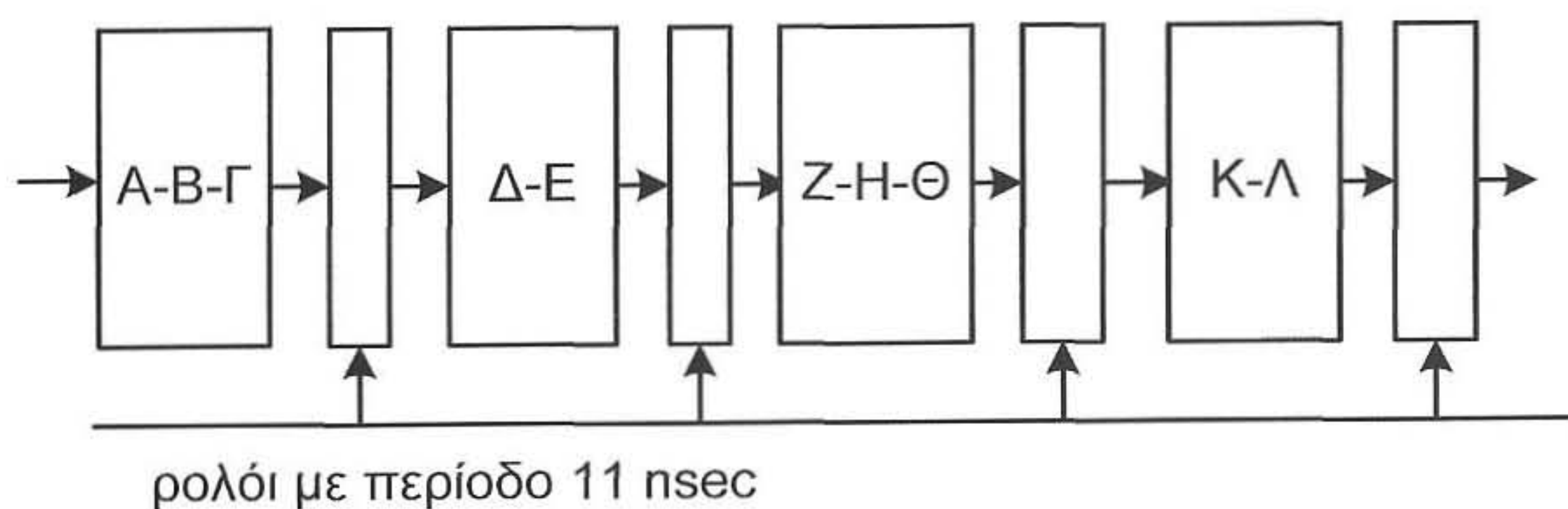
Η συνολική καθυστέρηση (latency) T του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών δίνεται από το γινόμενο του πλήθους των βαθμίδων του μηχανισμού και της περιόδου του ρολογιού. Λαμβάνοντας υπόψη ότι στη συγκεκριμένη περίπτωση όλες οι βαθμίδες δεν έχουν την ίδια καθυστέρηση, η συνολική καθυστέρηση T του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών θα είναι μεγαλύτερη του αθροίσματος της καθυστέρησης κάθε μονάδας και της καθυστέρησης που βάζει κάθε καταχωρητής του μηχανισμού. Δηλαδή $T \geq 3 + 4 + 3 + 5 + 3 + 4 + 3 + 3 + 5 + 4 + \lambda \times 1$ nsec, όπου λ είναι το πλήθος των καταχωρητών που χρησιμοποιούνται στο μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών (ή ισοδύναμα το πλήθος των βαθμίδων). Λαμβάνοντας υπόψη ότι σύμφωνα με την άσκηση πρέπει $T \leq 44$ nsec, από την παραπάνω σχέση λαμβάνουμε $3 + 4 + 3 + 5 + 3 + 4 + 3 + 3 + 5 + 4 + \lambda \times 1$ nsec $\leq T \leq 44$ nsec. Τότε $3 + 4 + 3 + 5 + 3 + 4 + 3 + 3 + 5 + 4 + \lambda \times 1$ nsec ≤ 44 nsec ή ισοδύναμα $37 + \lambda \times 1$ nsec ≤ 44 nsec, οπότε $\lambda \leq 44 - 37 = 7$. Επομένως πρέπει να έχουμε το πολύ 7 βαθμίδες. Επισημαίνουμε ότι όσο μειώνεται το πλήθος των βαθμίδων του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών, τόσο μειώνεται ο μέγιστος ρυθμός ολοκλήρωσης διαδικασιών.

Αφού η συνολική καθυστέρηση του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών δίνεται από το γινόμενο του πλήθους των βαθμίδων του μηχανισμού και της περιόδου του ρολογιού, αν είχαμε 7 βαθμίδες θα έπρεπε η καθυστέρηση κάθε βαθμίδας μαζί με τον καταχωρητή να είναι μικρότερη ή ίση του $\lfloor 44/7 \rfloor = 6$. Εάν προσπαθήσουμε να ομαδοποιήσουμε γειτονικές μονάδες, έτσι ώστε κάθε βαθμίδα να έχει μαζί με τον καταχωρητή καθυστέρηση το πολύ 6 nsec, τότε ο μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών θα έχει 10 βαθμίδες. Αυτή η λύση δεν είναι αποδεκτή διότι εμείς θεωρήσαμε ότι έχουμε 7 βαθμίδες.

Αν είχαμε 6 βαθμίδες θα έπρεπε η καθυστέρηση κάθε βαθμίδας μαζί με τον καταχωρητή να είναι μικρότερη ή ίση του $\lfloor 44/6 \rfloor = 7$. Εάν προσπαθήσουμε να ομαδοποιήσουμε γειτονικές μονάδες, έτσι ώστε κάθε βαθμίδα να έχει μαζί με τον καταχωρητή καθυστέρηση 7 nsec, τότε ο μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών θα έχει 9 βαθμίδες. Αυτή η λύση δεν είναι αποδεκτή διότι εμείς θεωρήσαμε ότι έχουμε 6 βαθμίδες.

Αν είχαμε 5 βαθμίδες θα έπρεπε η καθυστέρηση κάθε βαθμίδας μαζί με τον καταχωρητή να είναι μικρότερη ή ίση του $\lfloor 44/5 \rfloor = 8$. Εάν προσπαθήσουμε να ομαδοποιήσουμε γειτονικές μονάδες, έτσι ώστε κάθε βαθμίδα να έχει μαζί με τον καταχωρητή καθυστέρηση 8 nsec, τότε ο μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών θα έχει 7 βαθμίδες. Αυτή η λύση δεν είναι αποδεκτή διότι εμείς θεωρήσαμε ότι έχουμε 5 βαθμίδες.

Αν είχαμε 4 βαθμίδες θα έπρεπε η καθυστέρηση κάθε βαθμίδας μαζί με τον καταχωρητή να είναι μικρότερη ή ίση του $\lfloor 44/4 \rfloor = 11$. Εάν προσπαθήσουμε να ομαδοποιήσουμε γειτονικές μονάδες, έτσι ώστε κάθε βαθμίδα να έχει μαζί με τον καταχωρητή καθυστέρηση 11 nsec, τότε ο μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών θα έχει 4 βαθμίδες. Αυτή η λύση είναι αποδεκτή. Πράγματι οδηγεί σε συνολική καθυστέρηση του μηχανισμού ίση με $11 \times 4 \text{ nsec} = 44 \text{ nsec}$. Η απαιτούμενη ομαδοποίηση γειτονικών μονάδων σε βαθμίδες δίνεται στο σχήμα 4.4.1.



Σχήμα 4.4.1 Μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών.

Άσκηση 4.5 (4.3 βιβλίου)

Θεωρήστε δύο επεξεργαστές M1 και M2 με το ίδιο σύνολο εντολών. Οι εντολές διακρίνονται σε 4 κατηγορίες X_1 , X_2 , X_3 και X_4 . Κάθε μία εντολή αποτελείται από μια σειρά υπολειτουργιών που μπορούν να ομαδοποιηθούν σε 5 ομάδες και κάθε ομάδα εκτελείται σε μία από τις υπομονάδες ΠΕ, ΑΕ, ΕΕ, ΠΜ και ΑΑ. Ο χρόνος απασχόλησης κάθε υπομονάδας εξαρτάται από το είδος της εντολής και δίνεται στον Πίνακα 4.5.1.

Πίνακας 4.5.1

Κατηγορίες εντολών	ΠΕ	ΑΕ	ΕΕ	ΠΜ	ΑΑ
	ns	ns	ns	ns	ns
X_1	15	5	5	10	5
X_2	10	5	5	10	5
X_3	10	5	5	5	5
X_4	10	5	5	0	5

Ο επεξεργαστής M1 έχει ένα μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών (pipeline) με 5 βαθμίδες ΠΕ, ΑΕ, ΕΕ, ΠΜ και ΑΑ, που χρησιμοποιείται για την εκτέλεση όλων των εντολών. Ένας καταχωρητής μεταξύ δύο μονάδων του M1 βάζει καθυστέρηση 1 ns. Ο επεξεργαστής M2 δεν υλοποιεί την τεχνική μερικώς επικαλυπτόμενων λειτουργιών, χρησιμοποιεί όμως τις ίδιες υπομονάδες, η διάρκεια του κύκλου ρολογιού του είναι 5 ns και ολοκληρώνει κάθε εντολή σε όσους κύκλους ρολογιού απαιτούνται.

α. Να σχεδιάσετε το μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του M1 δίνοντας στο σχήμα όλη την απαιτούμενη πληροφορία.

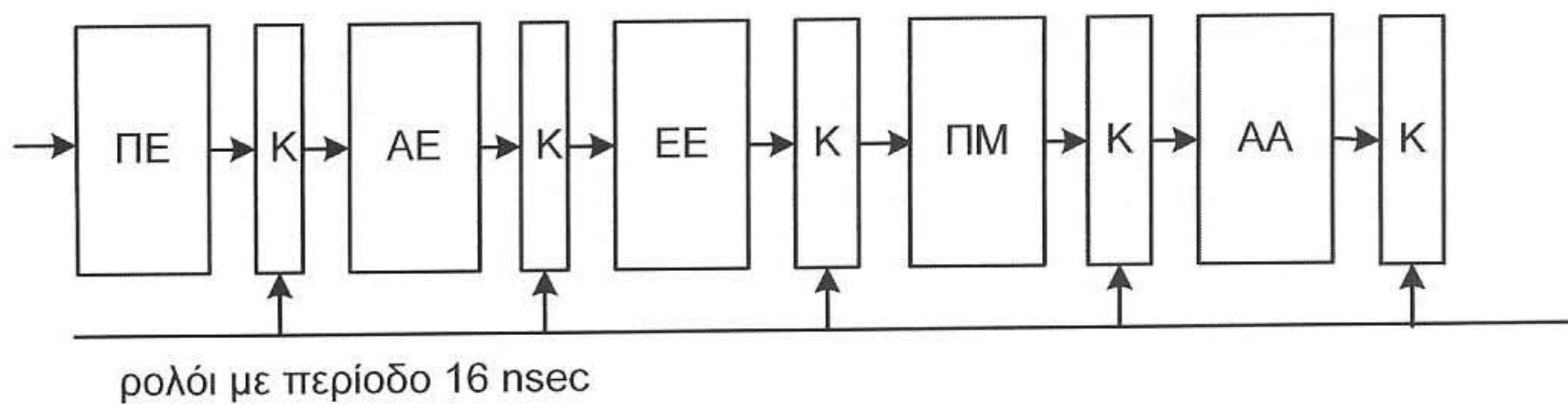
β. Να υπολογίσετε το χρόνο εκτέλεσης (latency) κάθε είδους εντολής στους επεξεργαστές M1 και M2.

γ. Να υπολογίσετε την επιτάχυνση του ρυθμού ολοκλήρωσης εντολών στην περίπτωση που

1. εκτελούνται 1000 εντολές του ίδιου είδους X_i με $i=1, 2, 3$, και 4.
2. εκτελούνται 1000 εντολές εκ των οποίων το 30% ανήκει στην κατηγορία X_1 , το 40% στην κατηγορία X_2 , το 20% στην κατηγορία X_3 και το 10% στην κατηγορία X_4 .

Απάντηση

α. Σ' ένα μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών η βαθμίδα με την μεγαλύτερη καθυστέρηση καθορίζει και την περίοδο του ρολογιού. Στην άσκησή μας για την εκτέλεση όλων των εντολών χρησιμοποιείται ένας μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών (pipeline) με 5 βαθμίδες ΠΕ, ΑΕ, ΕΕ, ΠΜ και ΑΑ,. Από τον Πίνακα 4.5.1 μπορούμε να δούμε ότι τη μεγαλύτερη καθυστέρηση βάζει η βαθμίδα ΠΕ στην περίπτωση εκτέλεσης εντολής του είδους X_1 , επομένως αυτή θα καθορίζει και την περίοδο του ρολογιού. Θα πρέπει όμως να λάβουμε υπόψη μας και την καθυστέρηση που βάζει ο καταχωρητής μεταξύ δύο βαθμίδων, οπότε η περίοδος του ρολογιού θα είναι $\tau = 15+1 \text{ nsec} = 16 \text{ nsec}$. Ο μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών δίνεται στο σχήμα 4.5.1.



Σχήμα 4.5.1 Μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών.

β. Στον επεξεργαστή M1 ο χρόνος εκτέλεσης $T(M1)$ είναι ίδιος για όλες τις εντολές:

$$T(M1) = \tau \times (\text{πλήθος βαθμίδων}) = 16 \times 5 = 80 \text{ nsec.}$$

Από τον Πίνακα παρατηρούμε ότι η καθυστέρηση σε κάθε μία από τις υπομονάδες ΠΕ, ΑΕ, ΕΕ, ΠΜ και ΑΑ είναι ακέραιο πολλαπλάσιο της περιόδου του ρολογιού, επομένως για να υπολογίσουμε το χρόνο εκτέλεσης μιας εντολής στον επεξεργαστή M2, αρκεί να προσθέσουμε την καθυστέρηση σε κάθε μία από τις υπομονάδες ΠΕ, ΑΕ, ΕΕ, ΠΜ και ΑΑ. Επομένως ο χρόνος εκτέλεσης μια εντολής της κατηγορίας i στον επεξεργαστή M2, $T_i(M2)$, θα είναι:

$$T_1(M2) = 15 + 5 + 5 + 10 + 5 = 40 \text{ nsec}$$

$$T_2(M2) = 10 + 5 + 5 + 10 + 5 = 35 \text{ nsec}$$

$$T_3(M2) = 10 + 5 + 5 + 5 + 5 = 30 \text{ nsec}$$

$$T_4(M2) = 10 + 5 + 5 + 0 + 5 = 25 \text{ nsec}$$

γ. Η επιτάχυνση του ρυθμού ολοκλήρωσης εντολών, EPO, είναι:

$$1. \text{ EPO}(X_1) = (40 \times 1000 \text{ nsec}) / [16 \times (4 + 1000) \text{ nsec}] = 40.000 / 16.064 = 2,49$$

$$\text{EPO}(X_2) = (35 \times 1000 \text{ nsec}) / [16 \times (4 + 1000) \text{ nsec}] = 35.000 / 16.064 = 2,17$$

$$\text{EPO}(X_3) = (30 \times 1000 \text{ nsec}) / [16 \times (4 + 1000) \text{ nsec}] = 30.000 / 16.064 = 1,87$$

$$\text{EPO}(X_4) = (25 \times 1000 \text{ nsec}) / [16 \times (4 + 1000) \text{ nsec}] = 25.000 / 16.064 = 1,56$$

$$2. \text{ EPO}(X_1) = [(40 \times 0,3 + 35 \times 0,4 + 30 \times 0,2 + 25 \times 0,1) \times 1000 \text{ nsec}] / [16 \times (4 + 1000) \text{ nsec}] = 34.500 / 16.064 = 2,15$$

Άσκηση 4.6 (4.4 βιβλίου)

α. Είναι δυνατόν να λυθεί η δομική εξάρτηση του επόμενου σχήματος (σχήμα 4.6 του βιβλίου) σε επίπεδο μεταγλωττιστή με την εισαγωγή της εντολής NOP μετά την εντολή SUB r3, r7, r6; Να δικαιολογήσετε την απάντησή σας.

εντολή	περίοδος σήματος χρονισμού								
	λ	λ+1	λ+2	λ+3	λ+4	λ+5	λ+6	λ+7	λ+8
LOAD r1, (r2)	ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ				
ADD r3, r4, r5		ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ			
SUB r3, r7, r6			ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ		
ADD r8, r4, r9				ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ	
AND r3, r4, r10					ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ

Σχήμα 4.6 του βιβλίου.

β. Να αναφέρετε τα πλεονεκτήματα και τα μειονεκτήματα της τεχνικής επίλυσης των εξαρτήσεων σε επίπεδο μεταγλωττιστή με εισαγωγή εντολών NOP σε σχέση με την τεχνική του κλειδώματος βαθμίδων του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών.

Απάντηση

α. Το πρόβλημα προκύπτει από το γεγονός ότι στο λ+3 κύκλο ρολογιού δεν μπορεί να προσπελαστεί η εντολή ADD r8, r4, r9, διότι έχουμε μία ενοποιημένη κρυφή μνήμη εντολών και δεδομένων η οποία προσπελάζεται ταυτόχρονα από τις βαθμίδες ΠΜ και ΠΕ. Επομένως εάν εισάγουμε μετά την εντολή SUB r3, r7, r6 μια εντολή NOP το πρόβλημα δεν λύνεται διότι και πάλι για την προσπέλαση της εντολής NOP απαιτείται η ταυτόχρονη προσπέλαση της ενοποιημένης κρυφής μνήμης από τις βαθμίδες ΠΜ και ΠΕ.

β. Η τεχνική κλειδώματος βαθμίδων του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών έχει το μειονέκτημα ότι απαιτεί επιπλέον κυκλώματα για την υλοποίησή της. Ένα πλεονέκτημά της είναι ότι μπορεί να χρησιμοποιηθεί και για την επίλυση δομικών εξαρτήσεων που αφορούν την ταυτόχρονη προσπέλαση ενοποιημένης κρυφής μνήμης εντολών και δεδομένων, ενώ η εντολή NOP δεν μπορεί να χρησιμοποιηθεί. Η επίλυση των εξαρτήσεων με χρήση εντολών NOP δεν απαιτεί επιπλέον κυκλώματα, αλλά οδηγεί σε αύξηση του πλήθους των εντολών του προγράμματος. Η αύξηση του πλήθους των εντολών του προγράμματος έχει ως συνέπεια την αύξηση του πλήθους των αποτυχιών

κατά την προσπέλαση της κρυφής μνήμης εντολών (ή της ενοποιημένης κρυφής μνήμης) και ως εκ τούτου αύξηση του μέσου χρόνου προσπέλασης εντολών, επομένως αύξηση του χρόνου εκτέλεσης του προγράμματος.

Άσκηση 4.7

Θεωρήστε το μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.4 του βιβλίου και ότι για την επίλυση των εξαρτήσεων χρησιμοποιείται η εντολή NOP. Θέλουμε να εκτελέσουμε τις επόμενες δύο εντολές κάποιας γλώσσας προγραμματισμού υψηλού επιπέδου:

$$A = B + C$$

$$D = E - F$$

Γράψτε σε συμβολική γλώσσα το πρόγραμμα που απαιτείται για την εκτέλεση των ανωτέρω δύο εντολών σε ένα υπολογιστή απλού συνόλου εντολών. Στη συνέχεια να αναδιατάξετε τις εντολές έτσι ώστε να έχουμε τη μεγαλύτερη απόδοση, χωρίς βέβαια να επηρεάζεται η σωστή εκτέλεση των εντολών.

Απάντηση

Αντικαθιστώντας τις εντολές $A = B + C$ και $D = E - F$ αντίστοιχα με εντολές συμβολικής γλώσσας λαμβάνουμε το πρόγραμμα 4.7.1.

Πρόγραμμα 4.7.1

LOAD r1, B	/ εντολή 1, $r1 \leftarrow B$
LOAD r2, C	/ εντολή 2
ADD r1, r2, r3	/ εντολή 3, $r3 \leftarrow r1 + r2$
STORE A, r3	/ εντολή 4, $A \leftarrow r3$
LOAD r4, E	/ εντολή 5
LOAD r5, F	/ εντολή 6
SUB r4, r5, r6	/ εντολή 7, $r6 \leftarrow r4 - r5$
STORE D, r6	/ εντολή 8

Στο σχήμα 4.7.1 δίνεται το χρονικό διάγραμμα χρήσης των βαθμίδων κατά την εκτέλεση του προγράμματος. Παρατηρούμε ότι η εντολή 3 χρησιμοποιεί το περιεχόμενο του καταχωρητή r2 στον οποίο όμως γράφει η προηγούμενη εντολή, η εντολή 2. Επομένως έχουμε μία εξάρτηση τύπου AME μεταξύ δύο διαδοχικών εντολών η οποία μπορεί να λυθεί παρεμβάλλοντας μεταξύ των εντολών 2 και 3 δύο εντολές NOP. Η εντολή 3 χρησιμοποιεί επίσης το περιεχόμενο του καταχωρητή r1 στον οποίο γράφει η εντολή 1. Λόγω όμως της παρεμβολής των δύο εντολών NOP μεταξύ των εντολών 2 και 3, μεταξύ των εντολών 1 και 3 παρεμβάλλονται συνολικά τρεις εντολές, επομένως δεν υπάρχει εξάρτηση μεταξύ των εντολών 1 και 3 (βλέπε πρόγραμμα 4.7.2). Η εντολή 3 γράφει στον καταχωρητή 3, ενώ η εντολή 4 διαβάζει από τον καταχωρητή 3, επομένως έχουμε εξάρτηση τύπου AME μεταξύ γειτονικών εντολών η οποία μπορεί να λυθεί παρεμβάλλοντας μεταξύ των εντολών 3 και 4 δύο εντολές NOP. Με τον ίδιο ακριβώς τρόπο μπορούν να λυθούν οι εξαρτήσεις μεταξύ των εντολών 6,7 και 7,8 (βλέπε πρόγραμμα 4.7.2).

Από την ανωτέρω συζήτηση προκύπτει ότι μεταξύ των δύο εντολών κάθε ενός από τα ζεύγη (2, 3), (3, 4), (6, 7) και (7, 8) πρέπει να παρεμβληθούν δύο ανεξάρτητες εντολές. Επομένως αναδιατάσσοντας τις εντολές με σκοπό τη χρήση του ελάχιστου αριθμού εντολών NOP παίρνουμε το πρόγραμμα 4.7.3.

εντολή	Περίοδος Σήματος Χρονισμού											
	λ	λ+1	λ+2	λ+3	λ+4	λ+5	λ+6	λ+7	λ+8	λ+9	λ+10	λ+11
LOAD r1, B	ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ							
LOAD r2, C		ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ						
ADD r1, r2, r3			ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ					
STORE A, r3				ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ				
LOAD r4, E					ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ			
LOAD r5, F						ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ		
SUB r4, r5, r6							ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ	
STORE D, r6								ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ

Σχήμα 4.7.1

Πρόγραμμα 4.7.2

LOAD r1, B	/ εντολή 1
LOAD r2, C	/ εντολή 2
NOP	
NOP	
ADD r1, r2, r3	/ εντολή 3
NOP	
NOP	
STORE A, r3	/ εντολή 4
LOAD r4, E	/ εντολή 5
LOAD r5, F	/ εντολή 6
NOP	
NOP	
SUB r4, r5, r6	/ εντολή 7
NOP	
NOP	
STORE D, r6	/ εντολή 8

Πρόγραμμα 4.7.3

LOAD r1, B
 LOAD r2, C
 LOAD r4, E
 LOAD r5, F
 ADD r1, r2, r3
 NOP
 SUB r4, r5, r6
 STORE A, r3
 NOP
 STORE D, r6

Άσκηση 4.8 (4.5 βιβλίου)

Θεωρήστε το μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.4 του βιβλίου και ότι έχει υλοποιηθεί η τεχνική παροχέτευσης για την επίλυση των εξαρτήσεων δεδομένων. Θέλουμε να εκτελέσουμε τις επόμενες δύο εντολές κάποιας γλώσσας προγραμματισμού υψηλού επιπέδου:

$$A = B + C$$

$$D = E - F$$

Γράψτε σε συμβολική γλώσσα το πρόγραμμα που απαιτείται για την εκτέλεση των ανωτέρω δύο εντολών σε ένα υπολογιστή απλού συνόλου εντολών. Στη συνέχεια να αναδιατάξετε τις εντολές έτσι ώστε να έχουμε τη μεγαλύτερη απόδοση, χωρίς βέβαια να επηρεάζεται η σωστή εκτέλεση του προγράμματος.

Απάντηση

Αντικαθιστώντας τις εντολές $A = B + C$ και $D = E - F$ αντίστοιχα με εντολές συμβολικής γλώσσας λαμβάνουμε το πρόγραμμα 4.8.1.

Πρόγραμμα 4.8.1

LOAD r1, B	/ εντολή 1, $r1 \leftarrow B$
LOAD r2, C	/ εντολή 2, $r2 \leftarrow C$
ADD r1, r2, r3	/ εντολή 3, $r3 \leftarrow r1 + r2$
STORE A, r3	/ εντολή 4, $A \leftarrow r3$
LOAD r4, E	/ εντολή 5, $r4 \leftarrow E$
LOAD r5, F	/ εντολή 6, $r5 \leftarrow F$
SUB r4, r5, r6	/ εντολή 7, $r6 \leftarrow r4 - r5$
STORE D, r6	/ εντολή 8, $D \leftarrow r6$

Όπως έχουμε δει, όταν στο μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.4 του βιβλίου υλοποιείται η τεχνική της παροχέτευσης, τότε παραμένουν εξαρτήσεις τύπου AME μόνον μεταξύ δύο γειτονικών εντολών εκ των οποίων η πρώτη είναι μια εντολή LOAD και η επόμενη χρησιμοποιεί τα δεδομένα που μεταφέρονται με την εντολή LOAD σε κάποιο καταχωρητή. Στο σχήμα 4.8.1 δίνεται το χρονικό διάγραμμα χρήσης των βαθμίδων κατά την εκτέλεση του προγράμματος και οι εξαρτήσεις που δεν επιλύονται με το μηχανισμό παροχέτευσης. Από το σχήμα αυτό μπορούμε να δούμε εύκολα ότι οι παραμένουσες εξαρτήσεις μπορούν να λυθούν εύκολα με τη χρήση δύο εντολών NOP, όπως φαίνεται στο πρόγραμμα 4.8.2.

εντολή	Περίοδος Σήματος Χρονισμού											
	λ	λ+1	λ+2	λ+3	λ+4	λ+5	λ+6	λ+7	λ+8	λ+9	λ+10	λ+11
LOAD r1, B	ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ							
LOAD r2, C		ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ						
ADD r1, r2, r3			ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ					
STORE A, r3				ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ				
LOAD r4, E					ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ			
LOAD r5, F						ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ		
SUB r4, r5, r6							ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ	
STORE D, r6								ΠΕ	ΑΕ	ΕΠ	ΠΜ	ΑΑ

Σχήμα 4.8.1 Χρονικό διάγραμμα χρήσης των βαθμίδων κατά την εκτέλεση του προγράμματος και εξαρτήσεις τύπου ΑΜΕ. Οι εξαρτήσεις δηλώνονται με τα ευθύγραμμα τμήματα.

Πρόγραμμα 4.8.2

LOAD r1, B	/ εντολή 1
LOAD r2, C	/ εντολή 2
NOP	
ADD r1, r2, r3	/ εντολή 3
STORE A, r3	/ εντολή 4
LOAD r4, E	/ εντολή 5
LOAD r5, F	/ εντολή 6
NOP	
SUB r4, r5, r6	/ εντολή 7
STORE D, r6	/ εντολή 8

Αναδιατάσσοντας τις εντολές παίρνουμε το πρόγραμμα 4.8.3 στο οποίο παρατηρούμε ότι δεν απαιτείται η χρησιμοποίηση εντολών NOP.

Πρόγραμμα 4.8.3

```
LOAD r1, B
LOAD r2, C
LOAD r4, E
LOAD r5, F
ADD r1, r2, r3
SUB r4, r5, r6
STORE A, r3
STORE D, r6
```

Ο μεταγλωττιστής, κατά τη μετάφραση των πηγαίων προγραμμάτων, αναδιατάσσει τις εντολές σε επίπεδο γλώσσας μηχανής, έτσι ώστε να ελαχιστοποιείται το πλήθος των εξαρτήσεων και να μειώνεται ο χρόνος εκτέλεσης του προγράμματος.

Άσκηση 4.9 (4.6 βιβλίου)

Στη συζήτηση που έγινε στο βιβλίο υποθέσαμε ότι οι κρυφές μνήμες εντολών και δεδομένων του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.4 είχαν όλες τις εντολές και τα δεδομένα αντίστοιχα που απαιτούνται. Τώρα θεωρήστε ότι:

- α. Ο λόγος επιτυχίας της κρυφής μνήμης εντολών και δεδομένων είναι 0,99 και 0,95 αντίστοιχα.
- β. Το 25% των εντολών είναι εντολές LOAD.
- γ. Κάθε αποτυχία συνεπάγεται κλείδωμα της βαθμίδας προσκόμισης εντολών για 40 χρονικές περιόδους (δηλαδή χάσιμο 40 χρονικών περιόδων).
- δ. Χρησιμοποιείται η τεχνική του κλειδώματος για την επίλυση κάθε είδους εξάρτησης.
- ε. Το 5% των εντολών του προγράμματος είναι εντολές BRE.
- ζ. Στο 1% των εντολών εμφανίζεται εξάρτηση τύπου AME μεταξύ γειτονικών εντολών και στο 2% μεταξύ μη γειτονικών εντολών.

Να υπολογίσετε το μέσο αριθμό κύκλων ρολογιού που απαιτείται για την εκτέλεση μιας εντολής.

Απάντηση

Εάν όλες οι εντολές βρίσκονταν στην κρυφή μνήμη εντολών, τα δεδομένα βρίσκονταν στην κρυφή μνήμη δεδομένων και δεν υπήρχαν εξαρτήσεις, τότε για την εκτέλεση 100 διαδοχικών εντολών ενός προγράμματος (όχι των πρώτων 100 διότι δεν μας ενδιαφέρουν οι αρχικές συνθήκες) θα απαιτούνταν 100 κύκλοι ρολογιού. Ωστόσο, στη συγκεκριμένη άσκηση, τίποτε από τα ανωτέρω δεν ισχύει.

Θα θεωρήσουμε ότι δεν συμβαίνει ταυτόχρονα στον ίδιο κύκλο ρολογιού αποτυχία στη βαθμίδα προσκόμισης εντολών (κρυφή μνήμη εντολών) και στη βαθμίδα προσπέλασης μνήμης (κρυφή μνήμη δεδομένων). Η θεώρηση αυτή είναι ρεαλιστική διότι οι λόγοι αποτυχίας της κρυφής μνήμης εντολών και της κρυφής μνήμης δεδομένων είναι πολύ μικροί. Στην περίπτωση της ταυτόχρονης στον ίδιο κύκλο ρολογιού αποτυχίας θα είχαμε καθυστέρηση μόνο 40 κύκλων ρολογιού.

Επειδή ο λόγος αποτυχίας της κρυφής μνήμης εντολών είναι $1 - 0,99 = 0,01$ και η αποτυχία συνεπάγεται κλείδωμα της βαθμίδας προσκόμισης εντολών για 40 χρονικές περιόδους, για την προσπέλαση 100 εντολών απαιτούνται $0,01 \times 100 \times 40 = 40$ επιπλέον κύκλοι ρολογιού.

Επειδή ο λόγος αποτυχίας της κρυφής μνήμης δεδομένων είναι $1 - 0,95 = 0,05$, το ποσοστό των εντολών που προσπελαύνουν δεδομένα, δηλαδή το ποσοστό των εντολών LOAD, είναι 25 % και η αποτυχία συνεπάγεται κλείδωμα της βαθμίδας προσκόμισης εντολών για 40 χρονικές περιόδους, για την προσπέλαση των δεδομένων απαιτούνται $0,05 \times 0,25 \times 100 \times 40 = 50$ επιπλέον κύκλοι ρολογιού.

Για κάθε εξάρτηση τύπου AME μεταξύ γειτονικών εντολών παγώνει (κλειδώνει) η βαθμίδα προσκόμισης νέας εντολής για δύο χρονικές περιόδους. Λαμβάνοντας υπόψη ότι στο 1% των εντολών εμφανίζεται εξάρτηση τύπου AME μεταξύ γειτονικών εντολών, συμπεραίνουμε ότι λόγω αυτών των εξαρτήσεων απαιτούνται $0,01 \times 100 \times 2 = 2$ επιπλέον χρονικές περίοδοι.

Για κάθε εξάρτηση τύπου AME μεταξύ μη γειτονικών εντολών (δηλαδή παρεμβάλλεται μία ανεξάρτητη εντολή) η βαθμίδα προσκόμισης νέας εντολής παγώνει για μία χρονική περίοδο. Λαμβάνοντας υπόψη ότι στο 2% των εντολών εμφανίζεται εξάρτηση τύπου AME μεταξύ μη γειτονικών εντολών, συμπαιρνούμε ότι λόγω αυτών των εξαρτήσεων απαιτούνται $0,02 \times 100 \times 1 = 2$ επιπλέον χρονικές περίοδοι.

Θεωρούμε ότι κάθε διαδικασιακή εξάρτηση λόγω μιας εντολής BRE οδηγεί για την επίλυσή της σε κλείδωμα της βαθμίδας προσκόμισης εντολών για δύο χρονικές περιόδους. Επομένως λόγω του ότι 5 % των εντολών είναι εντολές BRE, απαιτούνται $0,05 \times 100 \times 2 = 10$ επιπλέον χρονικές περίοδοι.

Επομένως για την εκτέλεση των 100 διαδοχικών εντολών του προγράμματος, αντί για 100 χρονικές περίοδοι, απαιτούνται $100 + 40 + 50 + 2 + 2 + 10 = 204$ χρονικές περίοδοι. Άρα κατά μέσο όρο κάθε $204/100 = 2,04$ χρονικές περιόδους ολοκληρώνεται η εκτέλεση και μιας νέας εντολής. Λαμβάνοντας υπόψη ότι ο μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών που θεωρήσαμε έχει 5 βαθμίδες ο μέσος αριθμός κύκλων ρολογιού που απαιτείται για την εκτέλεση μίας εντολής (καθυστερήση) είναι $5 \times 2,04$ κύκλοι ρολογιού = 10,2 κύκλοι ρολογιού.

Αν όλες οι εντολές και τα δεδομένα βρίσκονταν στην κρυφή μνήμη εντολών και δεδομένων αντίστοιχα και δεν υπήρχαν εξαρτήσεις, τότε σε κάθε κύκλο ρολογιού θα ολοκληρωνόταν η εκτέλεση και μιας εντολής, ενώ για την εκτέλεση μιας εντολής θα απαιτούνταν πέντε κύκλοι ρολογιού.

Άσκηση 4.10 (4.7 βιβλίου)

Θεωρήστε ότι οι εντολές BRE που εκτελούνται στο μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.4 του βιβλίου είναι εντολές καθυστερημένης διακλάδωσης με καθυστέρηση δύο χρονικών μονάδων. Να αναδιατάξετε τις εντολές του επόμενου προγράμματος, το οποίο γράφτηκε για υπολογιστή μη επικαλυπτόμενων λειτουργιών, έτσι ώστε να έχουμε τη μεγαλύτερη δυνατή απόδοση κατά την εκτέλεσή του στο μηχανισμό του σχήματος 4.4 του βιβλίου, χωρίς βέβαια να επηρεάζεται η σωστή εκτέλεση του προγράμματος. Θεωρήστε ότι ο υπολογιστής δεν υλοποιεί την τεχνική παροχέτευσης και ότι διαθέτει εντολές NOP.

...

```
SUB r5, r3, r7          // r7 ← r5-r3
AND r4, r5, r6
ADD r6, r4, r8
LOAD r1, (r3)           // r1 ← M(r3)
BRE r1, r2, 8H           // εάν r1-r2=0 τότε ΜΠ=ΜΠ+8(16)
STORE r4, (r1)          // M(r1) ← r4
```

.....

Απάντηση

Κατ' αρχήν ας δούμε τις τυχόν εξαρτήσεις που προκύπτουν κατά την εκτέλεση του ανωτέρω προγράμματος στο μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.4 του βιβλίου και τον τρόπο επίλυσής τους με την χρησιμοποίηση εντολών NOP.

Υπάρχει μια εξάρτηση τύπου AME μεταξύ των εντολών AND r4, r5, r6 και ADD r6, r4, r8 διότι η πρώτη γράφει στον r6 ενώ η δεύτερη διαβάζει τον r6. Για την επίλυση αυτής της εξάρτησης παρεμβάλουμε μεταξύ αυτών των δύο εντολών δύο εντολές NOP. Επίσης υπάρχει μια εξάρτηση τύπου AME μεταξύ των εντολών LOAD r1, (r3) και BRE r1, r2, 8H διότι η πρώτη γράφει στον r1 ενώ η δεύτερη διαβάζει τον r1. Για την επίλυση αυτής της εξάρτησης παρεμβάλουμε μεταξύ αυτών των δύο εντολών δύο εντολές NOP. Λαμβάνοντας υπόψη ότι η εντολή διακλάδωσης BRE είναι εντολή καθυστερημένης διακλάδωσης με καθυστέρηση δύο χρονικών μονάδων συμπεραίνουμε ότι οι δύο εντολές που έπονται θα εκτελεστούν άσχετα εάν ισχύει η συνθήκη της εντολής BRE ή όχι, επομένως προσθέτουμε μετά την εντολή BRE δύο εντολές NOP. Τότε θα προκύψει το πρόγραμμα 4.10.1:

Πρόγραμμα 4.10.1

.....

SUB r5, r3, r7

AND r4, r5, r6

NOP

NOP

ADD r6, r4, r8

LOAD r1, (r3)

NOP

NOP

BRE r1, r2, 8H

NOP

NOP

STORE r4, (r1)

.....

Στη συνέχεια θα προσπαθήσουμε να ελαχιστοποιήσουμε τη χρήση εντολών NOP. Στα προγράμματα 4.10.2 και 4.10.3 που δίνονται πιο κάτω παρατηρούμε ότι μεταξύ οποιωνδήποτε δύο εντολών με εξάρτηση τύπου AME παρεμβάλλονται δύο ανεξάρτητες εντολές. Επίσης μετά από την εντολή BRE υπάρχουν δύο εντολές που θα εκτελεστούν άσχετα εάν ικανοποιείται η συνθήκη της εντολής διακλάδωσης ή όχι. Και τα δύο προγράμματα είναι ισοδύναμα από πλευράς απόδοσης διότι και τα δύο χρησιμοποιούν τον ίδιο αριθμό εντολών NOP.

Πρόγραμμα 4.10.2

```
...
LOAD r1, (r3)
AND r4, r5, r6
SUB r5, r3, r7
BRE r1, r2, 8H
ADD r6, r4, r8
NOP
STORE r4, (r1)
.....
```

Πρόγραμμα 4.10.3

```
.....
LOAD r1, (r3)
AND r4, r5, r6
NOP
BRE r1, r2, 8H
SUB r5, r3, r7
ADD r6, r4, r8
STORE r4, (r1)
.....
```

Άσκηση 4.11

Θεωρήστε την Άσκηση 4.10 με μόνη διαφορά ότι ο υπολογιστής υλοποιεί την τεχνική παροχέτευσης.

Απάντηση

Κατ' αρχήν ας δούμε τις τυχόν εξαρτήσεις που προκύπτουν κατά την εκτέλεση του δοθέντος προγράμματος στο μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.4 του βιβλίου. Αφού ο υπολογιστής χρησιμοποιεί την τεχνική παροχέτευσης για την επίλυση εξαρτήσεων τύπου AME, θα έχουμε μόνο μια εξάρτηση μεταξύ των μεταξύ των εντολών LOAD r1, (r3) και BRE r1, r2, 8H διότι η πρώτη γράφει στον r1 ενώ η δεύτερη διαβάζει τον r1. Για την επίλυση αυτής της εξάρτησης πρέπει να παρεμβάλουμε μεταξύ αυτών των δύο εντολών μια εντολή NOP. Λαμβάνοντας υπόψη ότι η εντολή διακλάδωσης BRE είναι εντολή καθυστερημένης διακλάδωσης με καθυστέρηση δύο χρονικών μονάδων, προσθέτουμε προς το παρόν μετά την εντολή BRE δύο εντολές NOP. Τότε το πρόγραμμα που θα προκύψει είναι το κάτωθι:

```
...
SUB r5, r3, r7
AND r4, r5, r6
ADD r6, r4, r8
LOAD r1, (r3)
NOP
```



```
BRE r1, r2, 8H
NOP
NOP
STORE r4, (r1)
.....
```

Στη συνέχεια για να ελαχιστοποιήσουμε τη χρήση εντολών NOP μετακινούμε την εντολή SUB r5, r3, r7 μεταξύ των εντολών LOAD r1, (r3) και BRE r1, r2, 8H, οπότε η εντολή NOP δεν χρειάζεται πλέον. Επίσης μετακινούμε τις εντολές AND r4, r5, r6 και ADD r6, r4, r8 μετά την εντολή BRE, οπότε οι δύο εντολές NOP μετά την εντολή BRE δε χρειάζονται. Το πρόγραμμα που προέκυψε είναι:

```
...
LOAD r1, (r3)
SUB r5, r3, r7
BRE r1, r2, 8H
AND r4, r5, r6
ADD r6, r4, r8
STORE r4, (r1)
.....
```

Άσκηση 4.12

Θεωρήστε υπολογιστή απλού συνόλου εντολών με ξεχωριστή κρυφή μνήμη εντολών και δεδομένων, με μια εντολή των 32 δυαδικών ψηφίων ή ενός δεδομένου των 32 δυαδικών ψηφίων ανά θέση μνήμης και μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών των πέντε βαθμίδων: προσπέλασης εντολής (ΠΕ), αποκωδικοποίησης εντολής (ΑΕ), εκτέλεσης πράξης (ΕΠ), προσπέλασης μνήμης (ΠΜ) και αποθήκευσης αποτελέσματος (ΑΑ). Θεωρήστε επίσης ότι στη βαθμίδα ΑΕ ταυτόχρονα με την αποκωδικοποίηση της εντολής γίνεται και ανάγνωση καταχωρητών και ότι στον ίδιο κύκλο ρολογιού είναι δυνατόν να γίνει εγγραφή σε ένα καταχωρητή και ανάγνωση του περιεχομένου το οποίο γράφτηκε. Να υπολογίσετε το πλήθος των κύκλων ρολογιού που απαιτούνται για την εκτέλεση του τμήματος προγράμματος 4.12.1 σε συμβολική γλώσσα σε καθεμία των περιπτώσεων α και β. Και στις δύο περιπτώσεις να μην αλλάξετε τη σειρά εκτέλεσης των εντολών. Να δικαιολογήσετε τις απαντήσεις σας.

Πρόγραμμα 4.12.1

```

100  LOAD r0, #0          // r0 ← 0
101  LOAD r1, #9
102  LOAD r2, #1
103  LOAD r3, (r7)        // r3 ← M(r7)
104  LOAD r4, (r8)
105  ADD r3, r4, r5        // r3+r4 → r5
106  STORE r5, (r9)       // M(r9) ← r5
107  ADD r7, r2, r7
108  ADD r8, r2, r8
109  ADD r9, r2, r9
10A  SUB r1, r2, r1
10B  BRN r1, r0, -9H      // εάν r1-r0≠0 τότε ΜΠ=ΜΠ-9(16)
10C  END
    
```

α. Για την επίλυση κάθε είδους εξάρτησης χρησιμοποιείται η εντολή NOP.

β. Για την επίλυση εξαρτήσεων τύπου «ανάγνωση μετά από εγγραφή», ΑΜΕ, χρησιμοποιείται η τεχνική της παροχέτευσης. Για την επίλυση διαδικασιακών εξαρτήσεων χρησιμοποιείται η πρόβλεψη η οποία βασίζεται στην τιμή του αριθμού μετατόπισης (όταν ο αριθμός μετατόπισης είναι αρνητικός γίνεται η πρόβλεψη ότι θα εκτελεστεί η εντολή BRN).

Απάντηση

α.

Το τμήμα προγράμματος το οποίο δίνεται σ' αυτή την άσκηση δεν είναι δυνατόν να εκτελεστεί στο μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών της άσκησης λόγω των εξαρτήσεων που εμφανίζονται. Για την αντιμετώπιση των εξαρτήσεων θα χρησιμοποιήσουμε εντολές NOP. Για το μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών της άσκησης, ο οποίος είναι ίδιος με αυτόν του σχήματος 4.4 του βιβλίου, όταν μεταξύ δύο γειτονικών εντολών υπάρχει εξάρτηση τύπου ανάγνωσης μετά από εγγραφή θα πρέπει να παρεμβληθούν δύο εντολές NOP. Επίσης λαμβάνοντας υπόψη ότι η εντολή BRE ολοκληρώνεται στον τρίτο κύκλο, συνεπάγεται ότι για την επίλυση αυτής της διαδικασιακής εξάρτησης απαιτείται η εισαγωγή δύο εντολών NOP μετά την εντολή BRE. Με την κατάλληλη εισαγωγή εντολών NOP λαμβάνουμε το πρόγραμμα 4.12.2.

Πρέπει να σημειώσουμε ότι στο τμήμα προγράμματος 4.12.2, λόγω της παρεμβολής των εντολών NOP, έχει αλλάξει ο αριθμός μετατόπισης (displacement) της εντολής BRN από -9 σε -11 στο δεκαεξαδικό. Ο συμβολομεταφραστής θα αντικαταστήσει τον δεκαεξαδικό αριθμό -11 με την αναπαράστασή του σε συμπλήρωμα ως προς 2 κατά την παραγωγή του προγράμματος σε γλώσσα μηχανής.

Πρόγραμμα 4.12.2

```
100  LOAD r0, #0
101  LOAD r1, #9
102  LOAD r2, #1
103  LOAD r3, (r7)
104  LOAD r4, (r8)
105  NOP
106  NOP
107  ADD r3, r4, r5
108  NOP
109  NOP
10A  STORE r5, (r9)
10B  ADD r7, r2, r7
10C  ADD r8, r2, r8
10D  ADD r9, r2, r9
10E  SUB r1, r2, r1
10F  NOP
110  NOP
111  BRN r1, r0, -11H
112  NOP
113  NOP
114  END
```

Λαμβάνοντας υπόψη ότι έχουμε ένα μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών των πέντε βαθμίδων, συμπεραίνουμε ότι τους πρώτους 4 κύκλους ρολογιού δεν ολοκληρώνεται καμία εντολή. Τους επόμενους 3 κύκλους ρολογιού θα ολοκληρωθούν οι τρεις πρώτες εντολές του προγράμματος 4.12.2. Οι εντολές που είναι αποθηκευμένες στις θέσεις μνήμης από 103 έως και 113, ανήκουν σ' ένα βρόχο ο οποίος, λόγω των αρχικών περιεχομένων των καταχωρητών r1 και r2, θα εκτελεστεί 9 φορές. Επειδή αυτός ο βρόχος

περιλαμβάνει 17 εντολές, για την ολοκλήρωση αυτών των εντολών θα απαιτηθούν 17×9 κύκλοι ρολογιού. Θα απαιτηθεί επίσης, ένας επιπλέον κύκλος ρολογιού για την ολοκλήρωση της εντολής που είναι αποθηκευμένη στη θέση μνήμης με διεύθυνση 114. Επομένως ο συνολικός αριθμός κύκλων ρολογιού που θα απαιτηθούν για την εκτέλεση του προγράμματος 4.12.2 είναι:

$$4+3+17 \times 9+1=161 \text{ κύκλοι ρολογιού.}$$

β.

Σύμφωνα με την άσκηση για την επίλυση εξαρτήσεων τύπου «ανάγνωση μετά από εγγραφή», AME, χρησιμοποιείται η τεχνική της παροχέτευσης. Γνωρίζουμε ότι η τεχνική της παροχέτευσης δεν μπορεί να εξαλείψει πλήρως τις εξαρτήσεις που προκύπτουν μεταξύ ενός ζεύγους εντολών εκ των οποίων η πρώτη είναι μια εντολή LOAD. Σ' αυτή την περίπτωση απαιτείται η χρησιμοποίηση και μιας εντολής NOP. Επομένως μεταξύ των εντολών «LOAD r4, (r8)» και «ADD r3, r4, r5» του προγράμματος 4.12.1 θα εισάγουμε μια εντολή NOP. Σύμφωνα με την άσκηση για την επίλυση διαδικασιακών εξαρτήσεων χρησιμοποιείται η πρόβλεψη η οποία βασίζεται στην τιμή του αριθμού μετατόπισης. Επομένως λαμβάνουμε το τμήμα πρόγραμμα 4.12.3. Στο τμήμα προγράμματος 4.12.3, λόγω της παρεμβολής της εντολής NOP, έχει αλλάξει ο αριθμός μετατόπισης (displacement) της εντολής BRN από -9 σε -A στο δεκαεξαδικό.

Πρόγραμμα 4.12.3

```

100  LOAD r0, #0
101  LOAD r1, #9
102  LOAD r2, #1
103  LOAD r3, (r7)
104  LOAD r4, (r8)
105  NOP
106  ADD r3, r4, r5
107  STORE r5, (r9)
108  ADD r7, r2, r7
109  ADD r8, r2, r8
10A  ADD r9, r2, r9
10B  SUB r1, r2, r1
10C  BRN r1, r0, -AH
10D  END
    
```


Λαμβάνοντας υπόψη ότι έχουμε ένα μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών των πέντε βαθμίδων, συμπεραίνουμε ότι τους πρώτους 4 κύκλους ρολογιού δεν ολοκληρώνεται καμία εντολή. Τους επόμενους 3 κύκλους ρολογιού θα ολοκληρωθούν οι τρεις πρώτες εντολές του προγράμματος 4.12.3. Οι εντολές που είναι αποθηκευμένες στις θέσεις μνήμης από 103 έως και 10C, ανήκουν σ' ένα βρόχο ο οποίος, λόγω των αρχικών περιεχομένων των καταχωρητών r1 και r2, θα εκτελεστεί 9 φορές. Ωστόσο τις 8 πρώτες φορές εκτέλεσης της εντολής BRN θα γίνει σωστή πρόβλεψη, ενώ κατά την ένατη εκτέλεση της εντολής αυτής θα γίνει λάθος πρόβλεψη. Η λάθος πρόβλεψη γίνεται γνωστή στον τρίτο κύκλο της εντολής BRN, επομένως η εκτέλεση των εντολών, οι οποίες εκτελούνται στις βαθμίδες ΠΕ και ΑΕ του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών, σταματά και το αποτέλεσμα των λειτουργιών που έλαβαν χώρα σε δύο κύκλους ρολογιού μένει ανεκμετάλλευτο. Επειδή ο βρόχος περιλαμβάνει 10 εντολές, για την ολοκλήρωση αυτών των εντολών θα απαιτηθούν 10×9 κύκλοι ρολογιού, αλλά θα απαιτηθούν και δύο πρόσθετοι κύκλοι ρολογιού οι οποίοι χάνονται λόγω της λάθος πρόβλεψης κατά την ένατη εκτέλεση της εντολής BRN. Θα απαιτηθεί επίσης, ένας επιπλέον κύκλος ρολογιού για την ολοκλήρωση της εντολής που είναι αποθηκευμένη στη θέση μνήμης με διεύθυνση 10D. Επομένως ο συνολικός αριθμός κύκλων ρολογιού που θα απαιτηθούν για την εκτέλεση του προγράμματος 4.12.3 είναι:

$4+3+10 \times 9+2+1=100$ κύκλοι ρολογιού.

Συγκρίνοντας τους κύκλους του ρολογιού που απαιτούνται για την εκτέλεση του προγράμματος 4.12.2 σ' ένα μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών στο οποίο όλες οι εξαρτήσεις λύνονται με τη χρήση εντολών NOP, με τους κύκλους ρολογιού που απαιτούνται για την εκτέλεση του προγράμματος 4.12.3 σε ένα μηχανισμό που υλοποιεί την τεχνική της παροχέτευσης για την επίλυση εξαρτήσεων δεδομένων και την τεχνική της πρόβλεψης για την επίλυση διαδικασιακών εξαρτήσεων, παρατηρούμε ότι το πρόγραμμα εκτελείται σε πολύ λιγότερους κύκλους στο δεύτερο υπολογιστή.

Ασκηση 4.13

Θεωρήστε ότι για την εκτέλεση των εντολών ενός υπολογιστή απλού συνόλου εντολών του οποίου ο κύκλος ρολογιού περιγράφεται στην υποενότητα 4.1.3 του βιβλίου Αρχιτεκτονική Υπολογιστών, χρησιμοποιείται ο μηχανισμός μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.4 του ίδιου βιβλίου. Θεωρήστε επίσης ότι ο εν λόγω υπολογιστής διαθέτει πίνακα δακλαδώσεων (branch target buffer) με πεδίο ιστορίας διακλάδωσης (branch history) ισΔ του ενός δυαδικού ψηφίου.

α. Να περιγράψετε τι συμβαίνει κατά την προσπέλαση κάθε εντολής διακλάδωσης του προγράμματος που ακολουθεί και να δώσετε το περιεχόμενο που πίνακα διακλάδωσης πριν και μετά από κάθε αποτυχία πρόβλεψης. Θεωρήστε ότι οι εντολές που δεν δίνονται δεν είναι εντολές διακλάδωσης και είναι ανεξάρτητες από τις εντολές που δίνονται.

πρόγραμμα

```

.
.
.

0500  LOAD r1, #02H
0501  LOAD r2, #00H
.
.
.

1111  INC r2
1112  LOAD r4, #70H
1113  LOAD r5, #00H
.
.
.

1120  INC r5
.
.
.

1145  BRNE r4, r5, -26H      // εάν r4-r5≠0 τότε ΜΠ=ΜΠ-26(16)
.
.
.

1150  BRNE r1, r2, -40H
.
    
```


β. Να υπολογίσετε το ποσοστό των επιτυχημένων προβλέψεων αλλαγής της ροής του προγράμματος σε σχέση με το συνολικό αριθμό εκτέλεσης εντολών διακλάδωσης.

Απάντηση

Από το πρόγραμμα παρατηρούμε ότι υπάρχουν δύο βρόχοι, ο ένας εμφωλευμένος μέσα στον άλλο. Η πρώτη εντολή του εξωτερικού βρόχου είναι η εντολή «INC r2» και η τελευταία εντολή η εντολή «BRNE r1, r2, -40H». Ο εξωτερικός βρόχος θα εκτελεστεί δύο φορές. Η πρώτη εντολή του εσωτερικού βρόχου είναι η εντολή «INC r5» και η τελευταία είναι η εντολή «BRNE r4, r5, -26H». Για κάθε εκτέλεση του εξωτερικού βρόχου ο εσωτερικός βρόχος θα εκτελεστεί $70_{(16)} = 112_{(10)}$ φορές.

Θεωρούμε ότι σε κάθε θέση του πίνακα διακλαδώσεων υπάρχει ένα δυαδικό ψηφίο που καλείται δυαδικό ψηφίο εγκυρότητας και πριν την έναρξη της εκτέλεσης ενός προγράμματος όλα τα δυαδικά ψηφία εγκυρότητας του πίνακα διακλαδώσεων μηδενίζονται. Όταν για κάποια θέση του πίνακα διακλαδώσεων το δυαδικό ψηφίο εγκυρότητας έχει την τιμή 0, δεν μας ενδιαφέρει το περιεχόμενο των άλλων πεδίων αυτής της θέσης.

Στη βαθμίδα ΠΕ του μηχανισμού μερικώς επικαλυπτόμενων λειτουργιών προσπελάται η θέση της κρυφής μνήμης εντολών η διεύθυνση της οποίας βρίσκεται στον ΜΠ και ταυτόχρονα γίνεται προσπέλαση του πίνακα διακλαδώσεων με κλειδί το περιεχόμενο του ΜΠ. Εάν η θέση μνήμης με διεύθυνση το περιεχόμενο του ΜΠ δεν περιέχει μια εντολή διακλάδωσης, τότε προφανώς η προσπέλαση του πίνακα διακλαδώσεων θα είναι αποτυχία. Η βαθμίδα ΠΕ σε κάθε αποτυχημένη προσπάθεια προσπέλασης του πίνακα διακλαδώσεων θεωρεί ότι η εντολή που προσπελεύνεται δεν είναι εντολή διακλάδωσης. Στη συνέχεια θα ασχοληθούμε μόνο με τις εντολές διακλάδωσης.

Κατά την πρώτη προσπέλαση της εντολής «BRNE r4, r5, -26H» όλα τα δυαδικά ψηφία εγκυρότητας του πίνακα διακλαδώσεων θα έχουν την τιμή 0, το οποίο δηλώνει ότι ο πίνακας διακλαδώσεων δεν έχει χρήσιμο περιεχόμενο. Σ' αυτή την περίπτωση η βαθμίδα ΠΕ θεωρεί ότι η προσπέλαση δεν αφορά εντολή διακλάδωσης. Επομένως στο τέλος της τρέχουσας χρονικής περιόδου κατά τη μετάβαση του χρονικού σήματος από 0 σε 1 στον ΜΠ θα αποθηκευτεί η διεύθυνση της επόμενης εντολής του προγράμματος. Την επόμενη χρονική περίοδο, δεύτερος κύκλος ρολογιού της εντολής «BRNE r4, r5, -26 H», γίνεται αποκωδικοποίηση της εντολής και υπολογίζεται η διεύθυνση διακλάδωσης που καθορίζει η εντολή «BRNE r4, r5, -26 H». Αφού η εντολή είναι εντολή διακλάδωσης, το δυαδικό ψηφίο εγκυρότητας μιας θέσης (θεωρούμε της πρώτης) του πίνακα διακλαδώσεων θα πάρει την τιμή 1, στο πεδίο διεύθυνση της εντολής διακλάδωσης δΕΔ θα αποθηκευτεί η διεύθυνση της θέσης μνήμης που περιέχει την εν λόγω εντολή διακλάδωσης, δηλαδή το $1145_{(16)}$ και στο πεδίο διεύθυνση διακλάδωσης δΔ θα αποθηκευτεί η διεύθυνση της θέσης μνήμης η οποία περιέχει την επόμενη εντολή που πρέπει να εκτελεστεί εάν ισχύει η συνθήκη της εντολής διακλάδωσης, δηλαδή η διεύθυνση $1120_{(16)}$.

Την επόμενη χρονική περίοδο, τρίτος κύκλος εκτέλεσης της εντολής «BRNE r4, r5, -26H», ελέγχεται η συνθήκη διακλάδωσης της εντολής «BRNE r4, r5, -26H». Επειδή η συνθήκη διακλάδωσης ισχύει, στο τέλος αυτής της χρονικής περιόδου κατά την μετάβαση του σήματος χρονισμού από 0 σε 1 στον ΜΠ αποθηκεύεται η διεύθυνση διακλάδωσης που είναι 1120₍₁₆₎ και μηδενίζεται το περιεχόμενο των καταχωρητών κΑΕ και κΕΠ. Επομένως έχουμε χάσει δύο κύκλους ρολογιού. Επίσης στο πεδίο ισΔ της γραμμής του πίνακα διακλαδώσεων, η οποία στο πεδίο δΕΔ έχει την τιμή 1145, αποθηκεύεται η τιμή 1. Η τρέχουσα κατάσταση του πίνακα διακλαδώσεων δίνεται στο σχήμα 4.13.1.

Δυαδικό ψηφίο εγκυρότητας	δΕΔ	δΔ	ισΔ
1	1145	1120	1
0			
0			
.	.	.	.
.	.	.	.
.	.	.	.

Σχήμα 4.13.1

Κατά τη δεύτερη προσπέλαση της εντολής «BRNE r4, r5, -26H» γίνεται παράλληλα προσπέλαση του πίνακα διακλαδώσεων με κλειδί τον αριθμό 1145₍₁₆₎ ο οποίος είναι η διεύθυνση της θέσης μνήμης που περιέχει την εντολή «BRNE r4, r5, -26H». Από το σχήμα 4.13.1 παρατηρούμε ότι στον πίνακα διακλαδώσεων υπάρχει θέση στην οποία το δυαδικό ψηφίο του πεδίου εγκυρότητας έχει την τιμή 1 και το πεδίο δΕΔ την τιμή 1145. Επειδή η τιμή του πεδίου δΔ είναι 1120 συνεπάγεται ότι η διεύθυνση διακλάδωσης είναι το 1120 και επειδή η τιμή του πεδίου ισΔ έχει την τιμή 1, η βαθμίδα ΠΕ θα θεωρήσει ότι η συνθήκη διακλάδωσης θα ικανοποιηθεί. Επομένως στον επόμενο κύκλο ρολογιού θα προχωρήσει στην προσκόμιση της εντολής που βρίσκεται στη θέση μνήμης με διεύθυνση το περιεχόμενο του πεδίου δΔ, δηλαδή στη θέση μνήμης με διεύθυνση τον αριθμό 1120₍₁₆₎. Από το δοθέν πρόγραμμα διαπιστώνουμε ότι πράγματι ικανοποιείται η συνθήκη διακλάδωσης και η επόμενη προς εκτέλεση εντολή είναι η εντολή «INC r5» που είναι αποθηκευμένη στη θέση μνήμης με διεύθυνση το 1120₍₁₆₎. Το ίδιο θα γίνει και στις επόμενες 109₍₁₀₎ προσπελάσεις της εντολής «BRNE r4, r5, -26H». Κατά την 112 εκτέλεση του βρόχου, δηλαδή κατά την 112 προσπέλαση της εντολής «BRNE r4, r5, -26H», λόγω του περιεχομένου του πίνακα διακλαδώσεων και πάλι η βαθμίδα ΠΕ θα θεωρήσει ότι η συνθήκη της εντολής διακλάδωσης θα ικανοποιηθεί. Οπότε στον επόμενο κύκλο ρολογιού θα προχωρήσει στην προσπέλαση της εντολής που βρίσκεται αποθηκευμένη στη θέση μνήμης με διεύθυνση 1120₍₁₆₎. Ωστόσο δύο κύκλους ρολογιού αργότερα που

υπολογίζεται η συνθήκη διακλάδωσης διαπιστώνεται ότι η συνθήκη δεν ισχύει. Τότε μηδενίζεται το περιεχόμενο των καταχωρητών ΚΑΕ και ΚΕΠ και στον ΜΠ αποθηκεύεται η διεύθυνση της εντολής που έπεται της εντολής διακλάδωσης στο πρόγραμμα. Επομένως και σ' αυτή την περίπτωση έχουμε χάσει δύο κύκλους ρολογιού. Παράλληλα στον πίνακα διακλαδώσεων η τιμή του πεδίου ισΔ της γραμμής που στο πεδίο δΕΔ έχει την τιμή 1145₍₁₆₎ τίθεται ίση με 0. Η τρέχουσα κατάσταση του Πίνακα διακλαδώσεων δίνεται στο σχήμα 4.13.2.

Δυαδικό ψηφίο εγκυρότητας	δΕΔ	δΔ	ισΔ
1	1145	1120	0
0			
0			
.	.	.	.
.	.	.	.
.	.	.	.

Σχήμα 4.13.2

Κατά την πρώτη προσπέλαση της εντολής «BRNE r1, r2, -40H» προσπελάζεται παράλληλα ο πίνακας διακλαδώσεων με κλειδί τη διεύθυνση της θέσης μνήμης στην οποία είναι αποθηκευμένη η εντολή «BRNE r1, r2, -40H», δηλαδή τον αριθμό 1150₍₁₆₎. Επειδή δεν υπάρχει γραμμή του πίνακα διακλαδώσεων στην οποία το δυαδικό ψηφίο εγκυρότητας να έχει την τιμή 1 και το πεδίο δΕΔ να έχει την τιμή 1150, η βαθμίδα ΠΕ θεωρεί ότι η προσπέλαση δεν αφορά εντολή διακλάδωσης. Επομένως στο τέλος της τρέχουσας χρονικής περιόδου κατά τη μετάβαση του χρονικού σήματος από 0 σε 1 στον ΜΠ θα αποθηκευτεί η διεύθυνση της επόμενης εντολής του προγράμματος. Την επόμενη χρονική περίοδο, δεύτερος κύκλος ρολογιού της εντολής «BRNE r1, r2, -40H», γίνεται αποκωδικοποίηση της εντολής και υπολογίζεται η διεύθυνση διακλάδωσης που καθορίζει η εντολή «BRNE r1, r2, -40H». Αφού η εντολή είναι η εντολή διακλάδωσης, το δυαδικό ψηφίο εγκυρότητας μιας θέσης (θεωρούμε της δεύτερης) του πίνακα διακλαδώσεων θα πάρει την τιμή 1, στο πεδίο δΕΔ θα αποθηκευτεί η διεύθυνση της θέσης μνήμης που περιέχει την εν λόγω εντολή διακλάδωσης, δηλαδή το 1150₍₁₆₎ και στο πεδίο διεύθυνση διακλάδωσης δΔ θα αποθηκευτεί η διεύθυνση της θέσης μνήμης η οποία περιέχει την επόμενη εντολή που πρέπει να εκτελεστεί, δηλαδή η διεύθυνση 1111₍₁₆₎. Την επόμενη χρονική περίοδο, τρίτος κύκλος εκτέλεσης της εντολής «BRNE r1, r2, -40H», ελέγχεται η συνθήκη διακλάδωσης της εντολής «BRNE r1, r2, -40H». Επειδή η συνθήκη διακλάδωσης ισχύει, στο τέλος αυτής της χρονικής περιόδου κατά την μετάβαση του

σήματος χρονισμού από 0 σε 1 στον ΜΠ αποθηκεύεται η διεύθυνση διακλάδωσης που είναι $1111_{(16)}$ και μηδενίζεται το περιεχόμενο των καταχωρητών ΚΑΕ και ΚΕΠ. Επομένως έχουμε χάσει άλλους δύο κύκλους ρολογιού. Στο πεδίο $\text{ισ}\Delta$ της γραμμής του πίνακα διακλαδώσεων, η οποία στο πεδίο $\delta\text{Ε}\Delta$ έχει την τιμή 1150, αποθηκεύεται η τιμή 1. Η τρέχουσα κατάσταση του πίνακα διακλαδώσεων δίνεται στο σχήμα 4.13.3.

Δυαδικό ψηφίο εγκυρότητας	$\delta\text{Ε}\Delta$	$\delta\Delta$	$\text{ισ}\Delta$
1	1145	1120	0
1	1150	1111	1
0			
.	.	.	.
.	.	.	.
.	.	.	.

Σχήμα 4.13.3

Στη συνέχεια έχουμε πάλι προσπέλαση της εντολής «BRNE r4, r5, -26H». Παράλληλα γίνεται προσπέλαση του πίνακα διακλαδώσεων με κλειδί τη διεύθυνση της εντολής διακλάδωσης, δηλαδή το $1145_{(16)}$. Παρατηρούμε ότι υπάρχει γραμμή με δυαδικό ψηφίο εγκυρότητας που έχει την τιμή 1 και πεδίο $\delta\text{Ε}\Delta$ που έχει την τιμή 1145. Ωστόσο το πεδίο $\text{ισ}\Delta$ έχει την τιμή 0, επομένως, η βαθμίδα ΠΕ θα προβλέψει ότι η συνθήκη διακλάδωσης δεν θα ισχύσει, οπότε στον επόμενο κύκλο ρολογιού θα προσπελαστεί η εντολή που έπεται της εντολής διακλάδωσης στο πρόγραμμα. Όμως μετά από δύο χρονικές περιόδους που θα υπολογιστεί η συνθήκη διακλάδωσης της εντολής «BRNE r4, r5, -26H» θα διαπιστωθεί ότι ισχύει. Τότε θα μηδενιστεί το περιεχόμενο των καταχωρητών ΚΑΕ και ΚΕΠ και στον ΜΠ θα αποθηκευτεί η διεύθυνση διακλάδωσης, δηλαδή το $1120_{(16)}$ ώστε την επόμενη χρονική περίοδο να προσκομιστεί η σωστή εντολή. Επομένως έχουν χαθεί άλλοι δύο κύκλοι ρολογιού. Παράλληλα στον πίνακα διακλαδώσεων η τιμή του πεδίου $\text{ισ}\Delta$ της γραμμής η οποία στο πεδίο $\delta\text{Ε}\Delta$ έχει την τιμή $1145_{(16)}$, τίθεται στην τιμή 1. Το τρέχον περιεχόμενο του πίνακα διακλαδώσεων δίνεται στο σχήμα 4.13.4.

Δυαδικό ψηφίο εγκυρότητας	δεΔ	δΔ	ισΔ
1	1145	1120	1
1	1150	1111	1
0			
.	.	.	.
.	.	.	.
.	.	.	.

Σχήμα 4.13.4

Στις επόμενες 110 προσπελάσεις της εντολής «BRNE r4, r5, -26H» θα γίνει επιτυχής πρόβλεψη και το περιεχόμενο του πίνακα διακλαδώσεων δεν θα αλλάξει, θα είναι αυτό του σχήματος 4.13.4.

Κατά την επόμενη, 112^η προσπέλαση της εντολής «BRNE r4, r5, -26H» έχουμε αποτυχία πρόβλεψης, οπότε χάνονται δύο κύκλοι ρολογιού και η τιμή του πεδίου ισΔ λαμβάνει την τιμή 0. Η τρέχουσα κατάσταση του πίνακα διακλαδώσεων δίνεται στο σχήμα 4.13.5.

Δυαδικό ψηφίο εγκυρότητας	δεΔ	δΔ	ισΔ
1	1145	1120	0
1	1150	1111	1
0			
.	.	.	.
.	.	.	.
.	.	.	.

Σχήμα 4.13.5

Κατά την επόμενη, 2^η προσπέλαση της εντολής «BRNE r1, r2, -40H» έχουμε αποτυχία πρόβλεψης, οπότε χάνονται δύο κύκλοι ρολογιού και η τιμή του πεδίου ισΔ λαμβάνει την τιμή 0. Η τρέχουσα κατάσταση του πίνακα διακλαδώσεων δίνεται στο σχήμα 4.13.6.

Δυαδικό ψηφίο εγκυρότητας	δεΔ	δΔ	ισΔ
1	1145	1120	0
1	1150	1111	0
0			
.	.	.	.
.	.	.	.
.	.	.	.

Σχήμα 4.13.6

β. Οι εντολές διακλάδωσης εκτελούνται συνολικά $2 \times 112 + 2 = 226$ φορές. Από αυτές μόνο σε 6 εκτελέσεις έχουμε αποτυχία πρόβλεψης της επόμενης προς εκτέλεση εντολής. Επομένως το ποσοστό επιτυχημένης πρόβλεψης είναι $(226 - 6) / 226 = 97,3\%$.

Άσκηση 4.14 (4.8 βιβλίου)

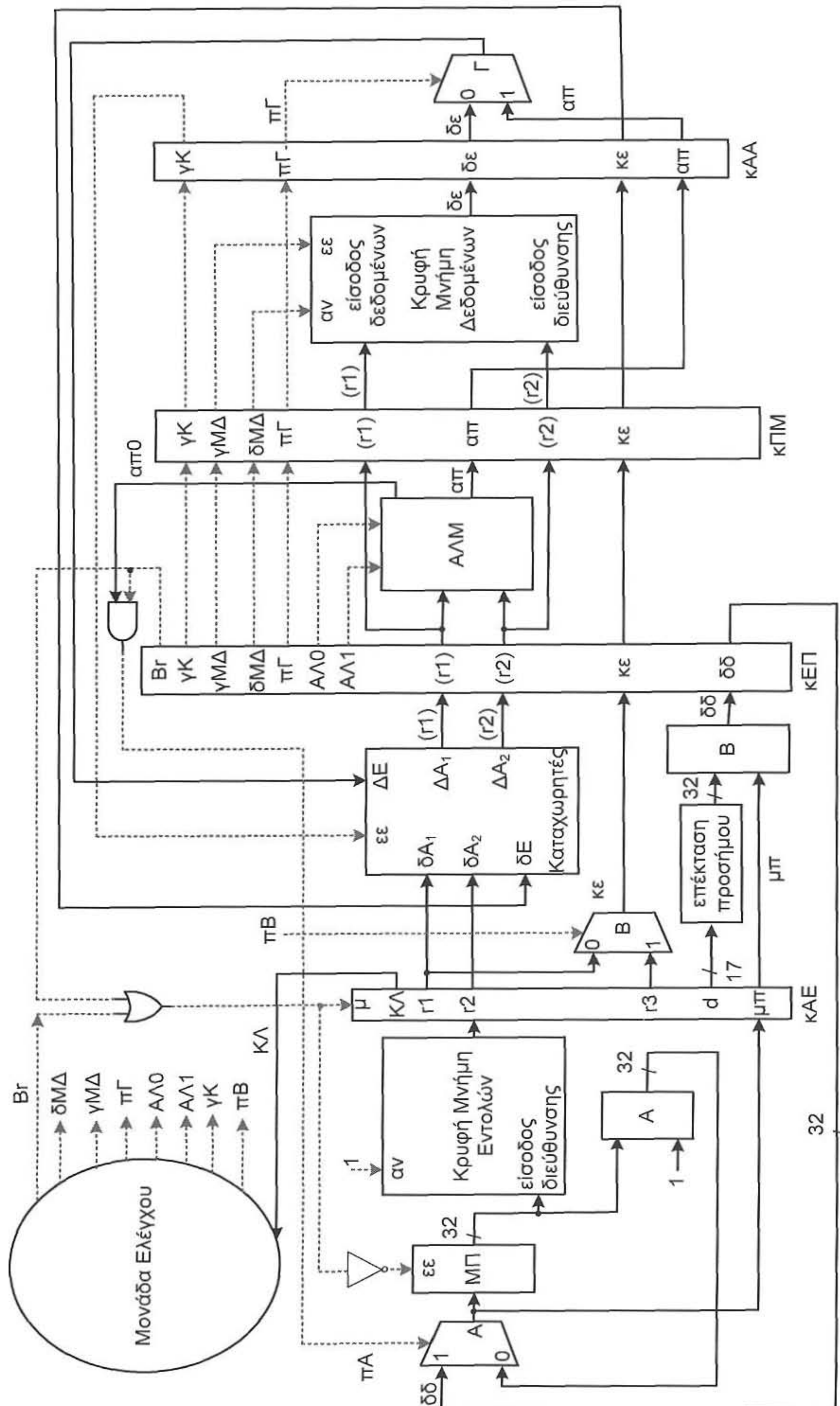
α. Πόσες εντολές NOP πρέπει να εισάγει ο μεταγλωττιστής μετά από κάθε εντολή BRE για την επίλυση της διαδικασιακής εξάρτησης που προκύπτει στο μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.42 του βιβλίου;

β. Να περιγράψετε λεπτομερειακά τις τροποποιήσεις και τα πρόσθετα κυκλώματα που απαιτούνται ώστε στο μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.42 οι διαδικασιακές εξαρτήσεις να μην επιλύονται σε επίπεδο μεταγλωττιστή αλλά με την τεχνική κλειδώματος βαθμίδων.

Απάντηση

α. Στο μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του σχήματος 4.42 του βιβλίου κατά την εκτέλεση μιας εντολής BRE η διεύθυνση διακλάδωσης υπολογίζεται στο δεύτερο κύκλο ρολογιού στη βαθμίδα ΑΕ ενώ ο έλεγχος της συνθήκης διακλάδωσης γίνεται στον τρίτο κύκλο ρολογιού στη βαθμίδα ΕΠ. Επομένως στο τέλος του τρίτου κύκλου ρολογιού κατά την μετάβαση του σήματος χρονισμού από μηδέν σε ένα στον ΜΠ αποθηκεύεται η διεύθυνση της επόμενης προς εκτέλεση εντολής. Άρα κατά τον δεύτερο και τον τρίτο κύκλο ρολογιού της εντολής BRE δεν πρέπει να προσκομίζεται νέα εντολή. Συνεπώς ο μεταγλωττιστής μετά από κάθε εντολή BRE πρέπει να εισάγει δύο εντολές NOP.

β. Όπως εξηγήσαμε στο παραπάνω ερώτημα κατά την εκτέλεση μιας εντολής BRE η διεύθυνση διακλάδωσης υπολογίζεται στην βαθμίδα AE, ενώ ο έλεγχος της συνθήκης γίνεται στη βαθμίδα ΕΠ. Επομένως όταν η εντολή BRE είναι στην βαθμίδα AE κατά την μετάβαση του σήματος χρονισμού από μηδέν σε ένα δεν πρέπει να αποθηκευτεί στον ΜΠ νέα τιμή. Αυτό επιτυγχάνεται εάν ο ΜΠ διαθέτει είσοδο ενεργοποίησης εγγραφής εε και όταν στη βαθμίδα AE βρίσκεται εντολή BRE η είσοδος αυτή εε του ΜΠ να οδηγείται με την ανενεργό τιμή, που έχουμε θεωρήσει ότι είναι το μηδέν. Όταν στη βαθμίδα AE βρίσκεται εντολή BRE θα πρέπει επίσης κατά τη μετάβαση του σήματος χρονισμού από μηδέν σε ένα να μηδενίζεται το περιεχόμενο του καταχωτή κΑΕ, διότι αυτό ισοδυναμεί με την προσπέλαση μιας εντολής NOP. Επομένως ο καταχωρητής κΑΕ πρέπει να έχει μία σύγχρονη είσοδο μηδενισμού μ. Όταν η εντολή BRE βρίσκεται στη βαθμίδα ΕΠ τότε κατά τη μετάβαση του σήματος χρονισμού από μηδέν σε ένα θα πρέπει και πάλι ο ΜΠ να μην ενημερωθεί και ο κΑΕ να μηδενιστεί. Αυτό επιτυγχάνεται και πάλι οδηγώντας με την τιμή μηδέν την είσοδο εε του ΜΠ και με την τιμή ένα την είσοδο μ του κΑΕ. Το ερώτημα είναι πως παράγονται οι τιμές των σημάτων που θα οδηγήσουν τις εισόδους εε του ΜΠ και μ του κΑΕ. Όταν η εντολή BRE βρίσκεται στη βαθμίδα AE, η μονάδα ελέγχου παράγει ένα σήμα $B_r=1$ το οποίο μπορεί να χρησιμοποιηθεί για να οδηγήσει την είσοδο μ του κΑΕ και το συμπλήρωμά του την είσοδο εε του ΜΠ. Στην επόμενη όμως χρονική περίοδο που η εντολή BRE βρίσκεται στη βαθμίδα ΕΠ στη βαθμίδα AE βρίσκεται η εντολή NOP. Επομένως το σήμα B_r θα έχει την τιμή μηδέν. Όπως έχουμε αναφέρει όλα τα σήματα ελέγχου που παράγονται κατά την αποκωδικοποίηση μιας εντολής στη βαθμίδα AE και θα χρειαστούν αργότερα, στο τέλος της χρονικής περιόδου αποθηκεύονται στον καταχωρητή κΕΠ. Άρα όταν η εντολή BRE βρίσκεται στη βαθμίδα ΕΠ για την οδήγηση των εισόδων εε του ΜΠ και μ του κΑΕ χρησιμοποιείται η τιμή του πεδίου B_r του κΕΠ. (βλέπε σχήμα 4.14.1.).



ΚΕΦΑΛΑΙΟ 5

Ασκήσεις για το σύστημα μνήμης

Άσκηση 5.1

Θεωρήστε μια μνήμη SDRAM με χρόνο αναζωογόνησης 64 ms, η οποία αποτελείται από 8K γραμμές κυψελίδων. Υποθέστε ότι χρειάζονται τέσσερις κύκλοι ρολογιού για την προσπέλαση μιας γραμμής για αναζωογόνηση. Εάν η συχνότητα του ρολογιού είναι 133 MHz να υπολογίσετε το ποσοστό του χρόνου που δαπανάται για την αναζωογόνηση της μνήμης, καθώς και το ποσοστό του χρόνου για το οποίο η μνήμη είναι διαθέσιμη για προσπελάσεις από την εφαρμογή.

Απάντηση

Αφού η συχνότητα του ρολογιού είναι 133 MHz, η περίοδος θα είναι $1/(133\text{MHz})=1/(133\times 10^6)\text{ sec}=1\times 10^{-6}/133\text{ sec}=1/133\text{ }\mu\text{sec}$.

Λαμβάνοντας υπόψη ότι έχουμε $8\text{K}=8\times 2^{10}=8.192$ γραμμές και ότι χρειάζονται τέσσερις κύκλοι ρολογιού για την προσπέλαση μιας γραμμής για αναζωογόνηση, συνεπάγεται ότι για την αναζωογόνηση όλης της μνήμης χρειάζονται $4\times 8.192=32.768$ κύκλοι ρολογιού ή ισοδύναμα $32.768\times 1/133\text{ }\mu\text{sec}=32.768/133\text{ }\mu\text{sec}=246,4\text{ }\mu\text{sec}$. Επομένως, το ποσοστό του χρόνου που δαπανάται για την αναζωογόνηση της μνήμης είναι $(246,4\text{ }\mu\text{sec})/(64\text{ ms})=0,00385$ ή ισοδύναμα 0,385%. Προφανώς το ποσοστό του χρόνου για το οποίο η μνήμη είναι διαθέσιμη για προσπελάσεις από την εφαρμογή είναι $100\% - 0,385\% = 99,615\%$.

Άσκηση 5.2 (5.1 βιβλίου)

Θεωρήστε επεξεργαστή με αρτηρία διευθύνσεων 16 δυαδικών ψηφίων. Διαθέτετε τα Ολοκληρωμένα Κυκλώματα (ΟΚ) μνήμης του Πίνακα 5.2.1, με οργάνωση μίας ψηφιολέξης (byte) ανά θέση μνήμης.

Πίνακας 5.2.1

Μνήμη	Χωρητικότητα (ψηφιολέξεις)
SRAM1	2 K
SRAM2	2 K
SRAM3	4 K
SRAM4	4 K
SRAM5	8 K
SRAM6	8 K

Χρησιμοποιώντας όλα τα OK που αναφέρονται στον πίνακα και ότι επιπλέον κυκλώματα σας χρειάζονται, να προτείνετε μια αρχιτεκτονική του συστήματος μνήμης, να την σχεδιάσετε και να δώσετε το πεδίο διευθύνσεων που αντιστοιχεί σε κάθε OK μνήμης. Να καλυφθούν όλες οι διευθύνσεις μνήμης από την αρχική (0000_{16}) έως εκείνη που ορίζεται από τη χωρητικότητα των OK που διαθέτετε. Θεωρήστε ότι το εύρος της αρτηρίας δεδομένων ισούται με το πλήθος των δυαδικών ψηφίων στα οποία αντιστοιχεί μια διεύθυνση του συστήματος μνήμης.

Απάντηση

Τα OK μνήμης που έχουν χωρητικότητα 2K ψηφιολέξεις $= 2 \times 2^{10}$ ψηφιολέξεις $= 2^{11}$ ψηφιολέξεις θα έχουν 11 εισόδους διεύθυνσης, αυτά που έχουν χωρητικότητα 4K ψηφιολέξεις $= 2^2 \times 2^{10}$ ψηφιολέξεις $= 2^{12}$ ψηφιολέξεις θα έχουν 12 εισόδους διεύθυνσης και αυτά που έχουν χωρητικότητα 8K ψηφιολέξεις $= 2^3 \times 2^{10}$ ψηφιολέξεις $= 2^{13}$ ψηφιολέξεις θα έχουν 13 εισόδους διεύθυνσης.

Η άσκηση μας δίνει την ελευθερία να σχεδιάσουμε σύστημα μνήμης υψηλής ή χαμηλής τάξης διαφύλωσης. Επίσης μας επιτρέπει να επιλέξουμε το πλήθος των δυαδικών ψηφίων ανά θέση του συστήματος μνήμης που θα σχεδιάσουμε. Θα εξετάσουμε όλους τους δυνατούς συνδυασμούς που μπορούμε να έχουμε με τα διαθέσιμα OK μνήμης.

A. Σχεδίαση συστήματος μνήμης υψηλής τάξης διαφύλωσης.

A.1 Σε κάθε ψηφιολέξη του συστήματος μνήμης αντιστοιχεί μια διεύθυνση.

Αφού τα διαθέσιμα ολοκληρωμένα κυκλώματα μνήμης έχουν οργάνωση μιας ψηφιολέξης ανά θέση μνήμης συνεπάγεται ότι σ' αυτή την περίπτωση κάθε φορά θα διευθυνσιοδοτείται μία θέση μνήμης μόνο ενός OK μνήμης.

Στον Πίνακα 5.2.2 δίνουμε τις διευθύνσεις που αντιστοιχούν στις θέσεις μνήμης κάθε OK μνήμης στην περίπτωση που τα τοποθετήσουμε στο χώρο διευθύνσεων του συστήματος μνήμης με τη σειρά που εμφανίζονται στον Πίνακα 5.2.1. Συμπεραίνουμε ότι οι εισοδοί διεύθυνσης των OK SRAM1 και SRAM2 πρέπει να οδηγούνται από τα 11 λιγότερο σημαντικά δυαδικά ψηφία $A_{10} \dots A_1 A_0$ της διεύθυνσης. Τα OK SRAM3 και SRAM4 πρέπει να οδηγούνται από τα 12 λιγότερο σημαντικά δυαδικά ψηφία $A_{11} \dots A_1 A_0$ της διεύθυνσης. Παρατηρούμε ότι οι διευθύνσεις που αντιστοιχούν στο OK SRAM5 καθώς και αυτές που αντιστοιχούν στο OK SRAM6 είναι 2^{13} το πλήθος, αλλά δεν αντιστοιχεί στην πρώτη θέση κάθε ολοκληρωμένου η διεύθυνση 0000000000000 και στην τελευταία θέση η διεύθυνση 111111111111. Αυτό οφείλεται στο γεγονός ότι το συνολικό πλήθος των διευθύνσεων που αντιστοιχούν στα OK SRAM1, SRAM2, SRAM3 και SRAM4 δεν είναι ακέραιο πολλαπλάσιο του 8 K. Τα OK SRAM5 και SRAM6 πρέπει να οδηγούνται από τα 13 λιγότερο σημαντικά δυαδικά ψηφία $A_{12} \dots A_1 A_0$ της διεύθυνσης. Ωστόσο για να αντιστοιχεί στην πρώτη θέση του OK SRAM5 η διεύθυνση 0000000000000 και στην τελευταία θέση η διεύθυνση 111111111111, θα πρέπει η πιο σημαντική είσοδος διεύθυνσης του OK SRAM5 να οδηγείται από το συμπλήρωμα του A_{12} . Το ίδιο ισχύει και για το OK SRAM6.

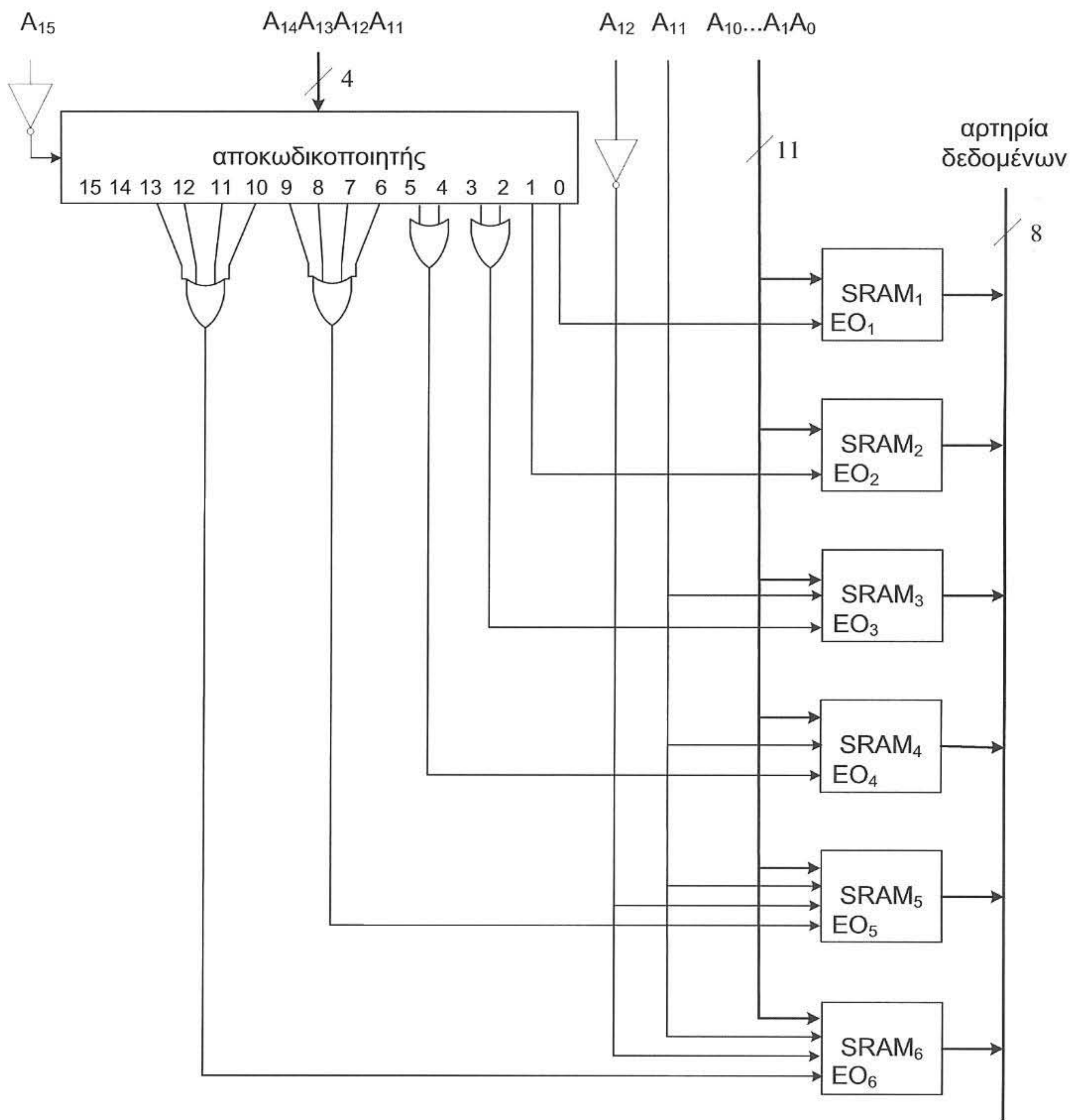
Πίνακας 5.2.2

Χάρτης διευθύνσεων του συστήματος μνήμης.

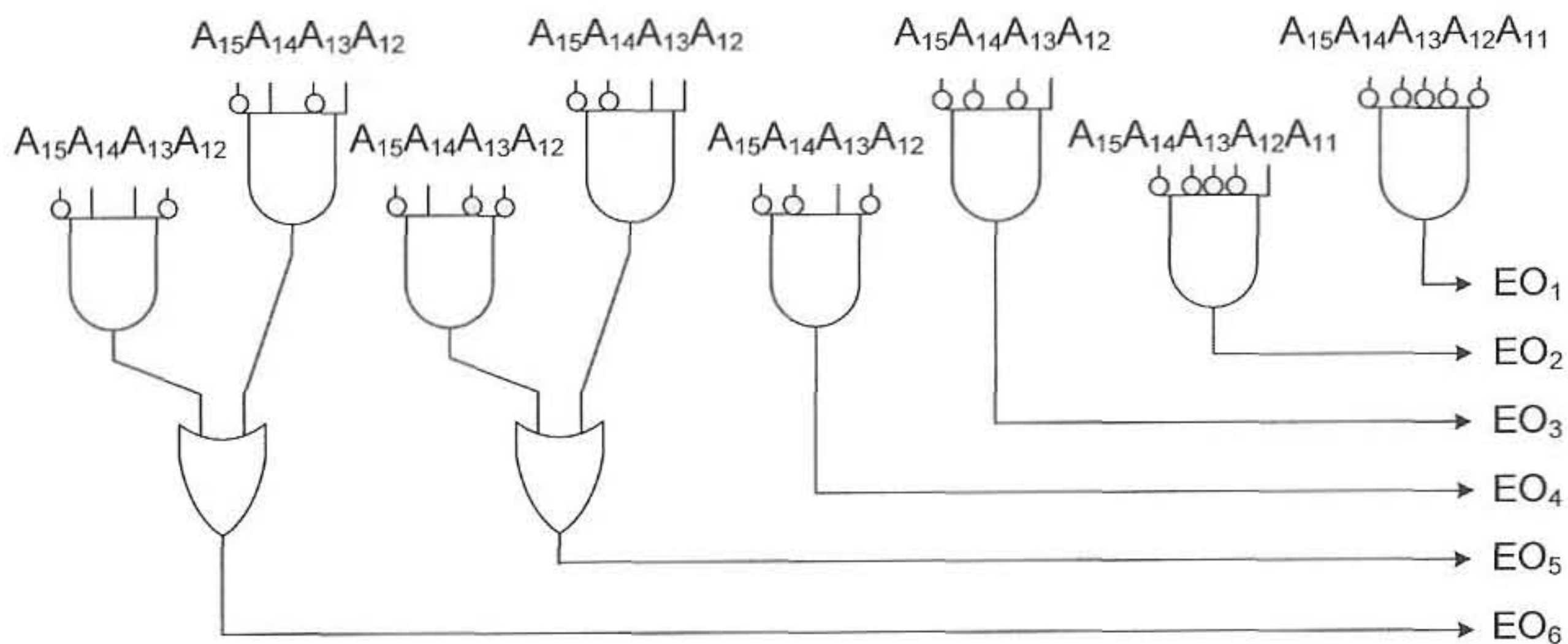
ολοκληρωμένο κύκλωμα μνήμης	πλήθος εισόδων διεύθυνσης	διευθύνσεις		
			A_{15}	\dots $A_1 A_0$
SRAM1	11	από	0000000000000000	
		έως	0000011111111111	
SRAM2	11	από	0000100000000000	
		έως	0000111111111111	
SRAM3	12	από	0001000000000000	
		έως	0001111111111111	
SRAM4	12	από	0010000000000000	
		έως	0010111111111111	
SRAM5	13	από	0011000000000000	
		έως	0011111111111111	
		και από	0100000000000000	
		έως	0100111111111111	
SRAM6	13	από	0101000000000000	
		έως	0101111111111111	
		και από	0110000000000000	
		έως	0110111111111111	

Παρατηρούμε ότι το πιο σημαντικό δυαδικό ψηφίο της διεύθυνσης είναι πάντα 0. Παρατηρούμε επίσης ότι το OK με την μικρότερη χωρητικότητα χρησιμοποιεί μόνο 11 από τις γραμμές διεύθυνσης. Επομένως για την παραγωγή των σημάτων ενεργοποίησης ολοκληρωμένου, ΕΟ, μπορούμε να χρησιμοποιήσουμε ένα αποκωδικοποιητή 4 σε 16 με εισόδους τα δυαδικά ψηφία A_{14} , A_{13} , A_{12} και A_{11} και είσοδο ενεργοποίησής του η οποία θα οδηγείται από το συμπλήρωμα του πιο σημαντικού ψηφίου A_{15} της διεύθυνσης. Από τον Πίνακα 5.2.2 παρατηρούμε ότι το OK SRAM1 ενεργοποιείται όταν $A_{14}A_{13}A_{12}A_{11}=0000$. Επομένως το σήμα ενεργοποίησης EO_1 του OK SRAM1 θα παράγεται από την έξοδο 0 του αποκωδικοποιητή. Παρατηρούμε ότι το OK SRAM2 ενεργοποιείται όταν $A_{14}A_{13}A_{12}A_{11}=0001$. Επομένως το σήμα ενεργοποίησης EO_2 του OK SRAM2 θα παράγεται από την έξοδο 1 του αποκωδικοποιητή. Παρατηρούμε ότι το OK SRAM3 ενεργοποιείται όταν $A_{14}A_{13}A_{12}=001$ και το $A_{11}=0$ ή 1. Επομένως το σήμα ενεργοποίησης EO_3 του SRAM3 θα παράγεται από μια πύλη OR με εισόδους τις εξόδους 2 και 3 του αποκωδικοποιητή. Το OK SRAM4 ενεργοποιείται όταν $A_{14}A_{13}A_{12}=010$ και το $A_{11}=0$ ή 1. Επομένως το σήμα ενεργοποίησης EO_4 του SRAM4 θα παράγεται από μια πύλη OR με εισόδους τις εξόδους 4 και 5 του αποκωδικοποιητή. Το OK SRAM5 ενεργοποιείται όταν $A_{14}A_{13}A_{12}A_{11}=0110$ ή 0111 ή 1000 ή 1001 . Επομένως το σήμα ενεργοποίησης EO_5 του SRAM5 θα παράγεται από μια πύλη OR με εισόδους τις εξόδους 6, 7, 8 και 9 του αποκωδικοποιητή. Το OK SRAM6 ενεργοποιείται όταν $A_{14}A_{13}A_{12}A_{11}=1010$ ή 1011 ή 1100 ή 1101 . Επομένως το σήμα ενεργοποίησης EO_6 του SRAM6 θα παράγεται από μια πύλη OR με εισόδους τις εξόδους 10, 11, 12 και 13 του αποκωδικοποιητή. Στο σχήμα 5.2.1 δίνεται το σύστημα της μνήμης.

Στο σχήμα 5.2.1 αντί της χρησιμοποίησης του αποκωδικοποιητή 4 σε 16, θα μπορούσαμε να παράγουμε τα σήματα ενεργοποίησης χρησιμοποιώντας 8 πύλες AND και 2 πύλες OR όπως φαίνεται στο σχήμα 5.2.2. Από τον Πίνακα 5.2.2 παρατηρούμε ότι οι τιμές των τριών πιο σημαντικών ψηφίων των διευθύνσεων που αντιστοιχούν στις πρώτες μισές διευθύνσεις του OK SRAM5 είναι ταυτοτικές με τις τιμές των τριών πιο σημαντικών ψηφίων των διευθύνσεων που αντιστοιχούν στις διευθύνσεις του SRAM4. Επίσης οι τιμές των τριών πιο σημαντικών ψηφίων των διευθύνσεων που αντιστοιχούν στις δεύτερες μισές διευθύνσεις του OK SRAM5 είναι ταυτοτικές με τις τιμές των τριών πιο σημαντικών ψηφίων των διευθύνσεων που αντιστοιχούν στις πρώτες μισές διευθύνσεις του OK SRAM6. Επομένως εάν για την παραγωγή των σημάτων ενεργοποίησης των OK SRAM5 και SRAM6 χρησιμοποιούσαμε μόνο τα τρία πιο σημαντικά δυαδικά ψηφία της διεύθυνσης θα ενεργοποιούνταν ταυτόχρονα σε κάποιες περιπτώσεις τα OK SRAM5 και SRAM4 και σε κάποιες άλλες τα SRAM5 και SRAM6. Το πρόβλημα λύνεται εάν χρησιμοποιήσουμε τα τέσσερα πιο σημαντικά δυαδικά ψηφία της διεύθυνσης, όπως φαίνεται στο σχήμα 5.2.2.



Σχήμα 5.2.1 Μονάδα ημιαγωγικής μνήμης 28 Kψηφιολέξεων με μία ψηφιολέξη ανά θέση μνήμης. Οι άλλες είσοδοι των ολοκληρωμένων κυκλωμάτων μνήμης, όπως ΕΕ και Εγγ., δεν σχεδιάστηκαν στο σχήμα διότι δεν επηρεάζουν τη διαμόρφωση της μονάδας μνήμης.



Σχήμα 5.2.2 Παραγωγή των σημάτων ενεργοποίησης με πύλες AND και OR.

Στη συνέχεια θα δούμε μια εναλλακτική σχεδίαση του συστήματος μνήμης.

Αφού η άσκηση δεν μας ζητάει να τοποθετήσουμε στο χώρο διευθύνσεων του συστήματος μνήμης τα OK με τη σειρά που δίνονται στον Πίνακα 5.2.1, μπορούμε να αλλάξουμε τη σειρά, έτσι ώστε η διεύθυνση που αντιστοιχεί στην πρώτη θέση κάθε OK μνήμης να είναι ακέραιο πολλαπλάσιο της χωρητικότητας του συγκεκριμένου OK (δηλαδή, για ένα OK με χωρητικότητα 2^k θέσεων η διεύθυνση που αντιστοιχεί στην πρώτη θέση του να έχει μηδενικά τα k λιγότερο σημαντικά της δυαδικά ψηφία).

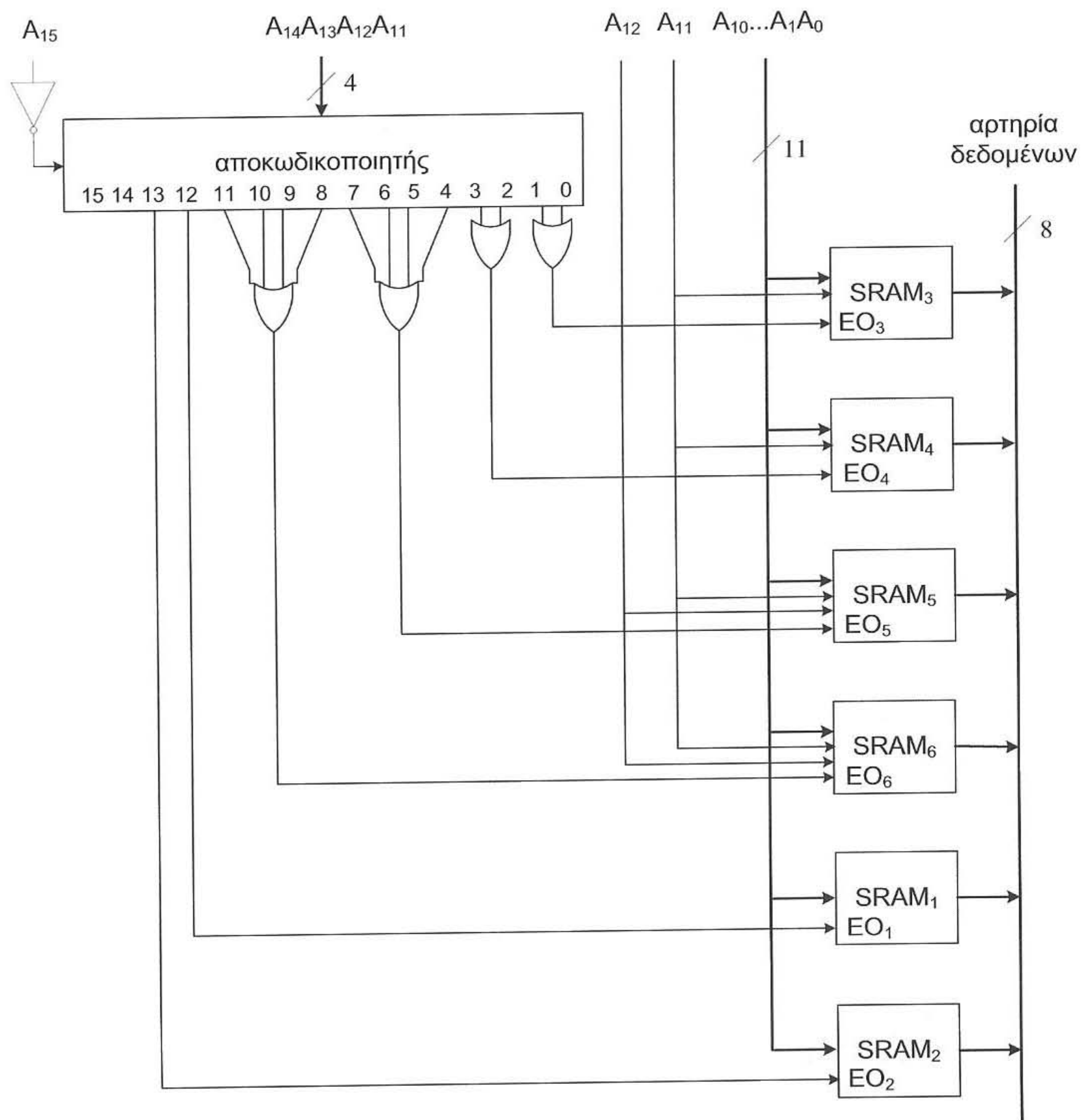
Πίνακας 5.2.3

Χάρτης διευθύνσεων του συστήματος μνήμης.

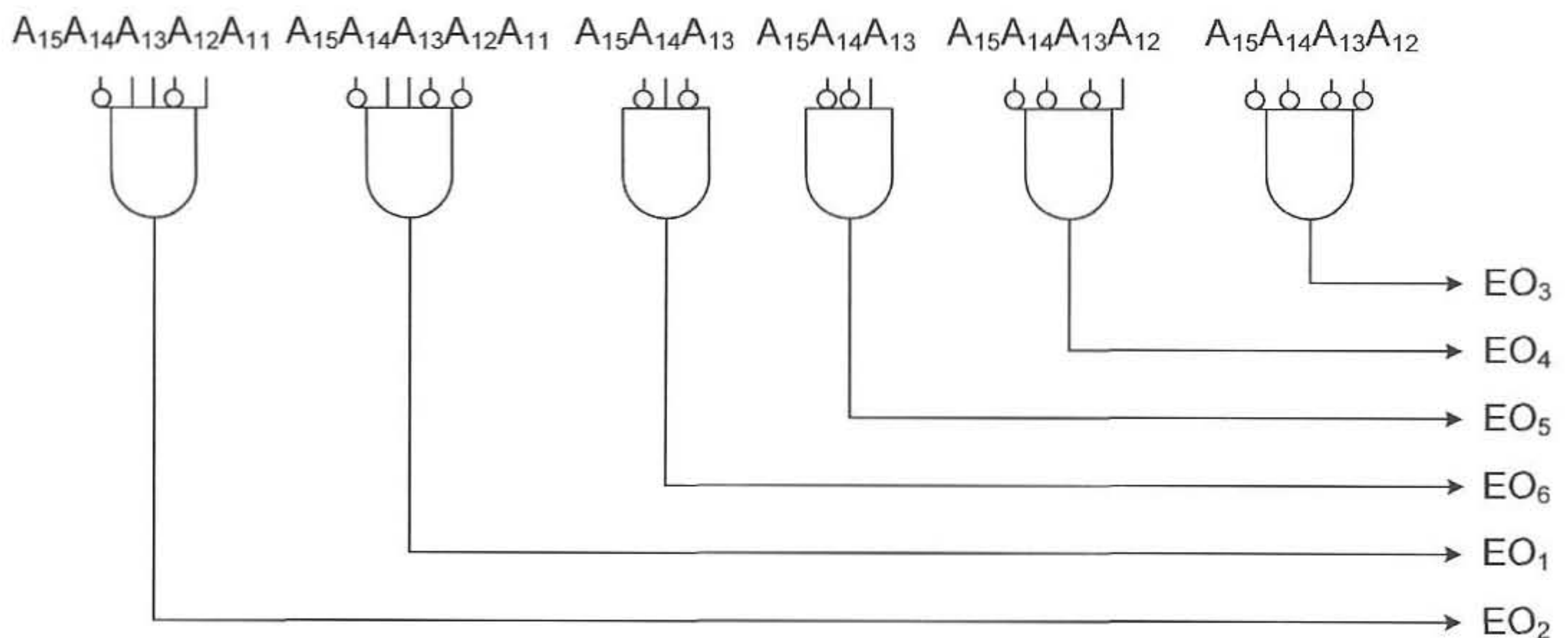
ολοκληρωμένο κύκλωμα μνήμης	πλήθος εισόδων διεύθυνσης	Διευθύνσεις		
		A_{15}	...	$A_1 A_0$
SRAM3	12	από	0000000000000000	
		έως	0000111111111111	
SRAM4	12	από	0001000000000000	
		έως	0001111111111111	
SRAM5	13	από	0010000000000000	
		έως	0011111111111111	
SRAM6	13	από	0100000000000000	
		έως	0101111111111111	
SRAM1	11	από	0110000000000000	
		έως	0110011111111111	
SRAM2	11	από	0110100000000000	
		έως	0110111111111111	

Υπάρχουν διάφοροι συνδυασμοί που ικανοποιούν την παραπάνω συνθήκη, στον Πίνακα 5.2.3 δίνεται ο χάρτης διευθύνσεων που αντιστοιχεί σε ένα τέτοιο συνδυασμό. Παρατηρούμε ότι το πιο σημαντικό δυαδικό ψηφίο της διεύθυνσης είναι πάντα 0. Παρατηρούμε επίσης ότι το OK με την μικρότερη χωρητικότητα χρησιμοποιεί μόνο 11 από τις γραμμές διεύθυνσης. Επομένως για την παραγωγή των σημάτων ενεργοποίησης ολοκληρωμένου, ΕΟ, μπορούμε και πάλι να χρησιμοποιήσουμε ένα αποκωδικοποιητή 4 σε 16 με εισόδους τα δυαδικά ψηφία A_{14} , A_{13} , A_{12} και A_{11} και είσοδο ενεργοποίησής του η οποία θα οδηγείται από το συμπλήρωμα του πιο σημαντικού ψηφίου A_{15} της διεύθυνσης. Από τον Πίνακα 5.2.3 παρατηρούμε ότι το OK SRAM3 ενεργοποιείται όταν $A_{14}A_{13}A_{12} = 000$ και $A_{11} = 0$ ή 1. Επομένως το σήμα ενεργοποίησης EO_3 του SRAM3 θα παράγεται από μια πύλη OR με εισόδους τις εξόδους 0 και 1 του αποκωδικοποιητή. Παρατηρούμε ότι το OK SRAM4 ενεργοποιείται όταν $A_{14}A_{13}A_{12} = 001$ και $A_{11} = 0$ ή 1. Επομένως το σήμα ενεργοποίησης EO_4 του SRAM4 θα παράγεται από μια πύλη OR με εισόδους τις εξόδους 2 και 3 του αποκωδικοποιητή. Παρατηρούμε ότι το OK SRAM5 ενεργοποιείται όταν $A_{14}A_{13} = 01$ και τα $A_{12}A_{11} = 00, 01, 10$ ή 11. Επομένως το σήμα ενεργοποίησης EO_5 του SRAM5 θα παράγεται από μια πύλη OR με εισόδους τις εξόδους 4, 5, 6 και 7 του αποκωδικοποιητή. Το OK SRAM6 ενεργοποιείται όταν $A_{14}A_{13} = 10$ και τα $A_{12}A_{11} = 00, 01, 10$ ή 11. Επομένως το σήμα ενεργοποίησης EO_6 του SRAM6 θα παράγεται από μια πύλη OR με εισόδους τις εξόδους 8, 9, 10 και 11 του αποκωδικοποιητή. Το OK SRAM1 ενεργοποιείται όταν $A_{14}A_{13}A_{12}A_{11} = 1100$. Επομένως το σήμα ενεργοποίησης EO_1 του SRAM1 θα παράγεται από την έξοδο 12 του αποκωδικοποιητή. Τελικά το OK SRAM2 ενεργοποιείται όταν $A_{14}A_{13}A_{12}A_{11} = 1101$. Επομένως το σήμα ενεργοποίησης EO_2 του SRAM2 θα παράγεται από την έξοδο 13 του αποκωδικοποιητή. Στο σχήμα 5.2.3 δίνεται το σύστημα της μνήμης.

Στο σχήμα 5.2.3 αντί της χρησιμοποίησης του αποκωδικοποιητή 4 σε 16 θα μπορούσαμε να παράγουμε τα σήματα ενεργοποίησης χρησιμοποιώντας 6 πύλες AND όπως φαίνεται στο σχήμα 5.2.4.



Σχήμα 5.2.3 Εναλλακτική σχεδίαση μονάδας ημιαγωγικής μνήμης 28 Κψηφιολέξεων με μία ψηφιολέξη ανά θέση μνήμης.



Σχήμα 5.2.4 Παραγωγή των σημάτων ενεργοποίησης με πύλες AND.

A.2 Σε κάθε δύο ψηφιολέξεις του συστήματος μνήμης αντιστοιχεί μια διεύθυνση.

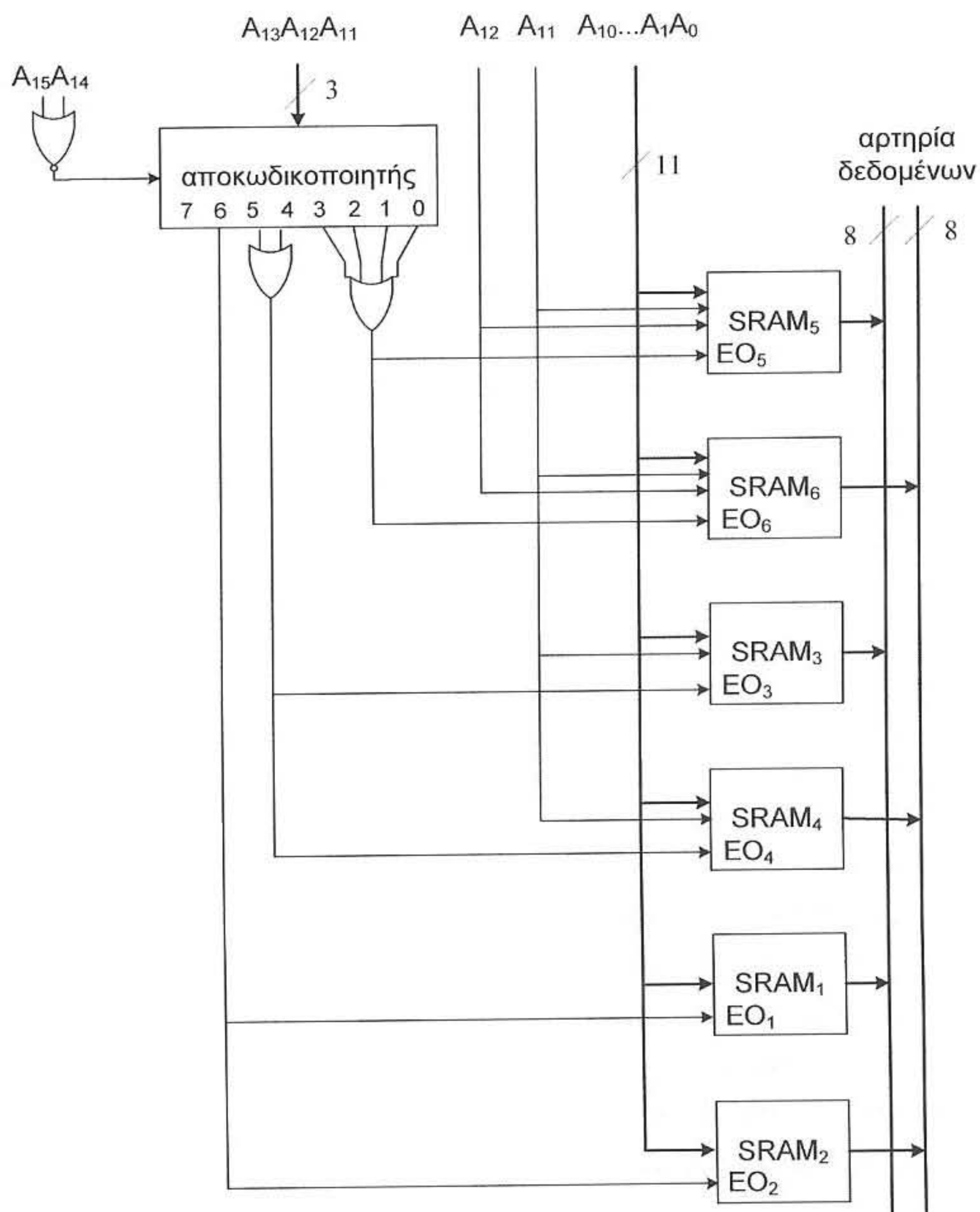
Για να έχουμε δύο ψηφιολέξεις ανά θέση μνήμης, θα πρέπει η ίδια διεύθυνση να αντιστοιχεί σε δύο θέσεις δύο διακριτών OK μνήμης. Είναι προφανές ότι η ίδια διεύθυνση δεν είναι δυνατόν να αντιστοιχεί σε δύο θέσεις του ίδιου OK μνήμης. Αφού έχουμε 2 OK μνήμης των 2 Kψηφιολέξεων, 2 OK μνήμης των 4 Kψηφιολέξεων και 2 OK μνήμης των 8 Kψηφιολέξεων θα σχεδιάσουμε το σύστημα μνήμης έτσι ώστε τα ζεύγη των OK που αναφέραμε να διευθυνσιοδοτούνται ταυτόχρονα (δηλαδή οι είσοδοι διεύθυνσης των OK να οδηγούνται από τις ίδιες γραμμές διευθύνσεων) και οι είσοδοι ενεργοποίησης OK να ενεργοποιούνται ταυτόχρονα.

Στον Πίνακα 5.2.4 τοποθετήσαμε τα OK στο χώρο διευθύνσεων του συστήματος μνήμης έτσι ώστε η διεύθυνση που αντιστοιχεί στην πρώτη θέση κάθε OK μνήμης να είναι ακέραιο πολλαπλάσιο της χωρητικότητας του συγκεκριμένου OK. Από τον Πίνακα 5.2.4 παρατηρούμε ότι τα δύο πιο σημαντικά δυαδικά ψηφία των διευθύνσεων έχουν πάντα τη λογική τιμή 0. Επομένως στην περίπτωση αυτή για να παράγουμε τα σήματα ενεργοποίησης των OK μνήμης μπορούμε να χρησιμοποιήσουμε αποκωδικοποιητή 3 σε 8 με εισόδους $A_{13}A_{12}A_{11}$ και είσοδο ενεργοποίησης η οποία οδηγείται από την έξοδο μιας πύλης NOR με εισόδους A_{15} και A_{14} . Στο σχήμα 5.2.5 δίνεται το σύστημα της μνήμης που αντιστοιχεί στο χάρτη διευθύνσεων του Πίνακα 5.2.4. Αντί της χρησιμοποίησης του αποκωδικοποιητή 3 σε 8 θα μπορούσαμε να παράγουμε τα σήματα ενεργοποίησης χρησιμοποιώντας 3 πύλες AND όπως φαίνεται στο σχήμα 5.2.6.

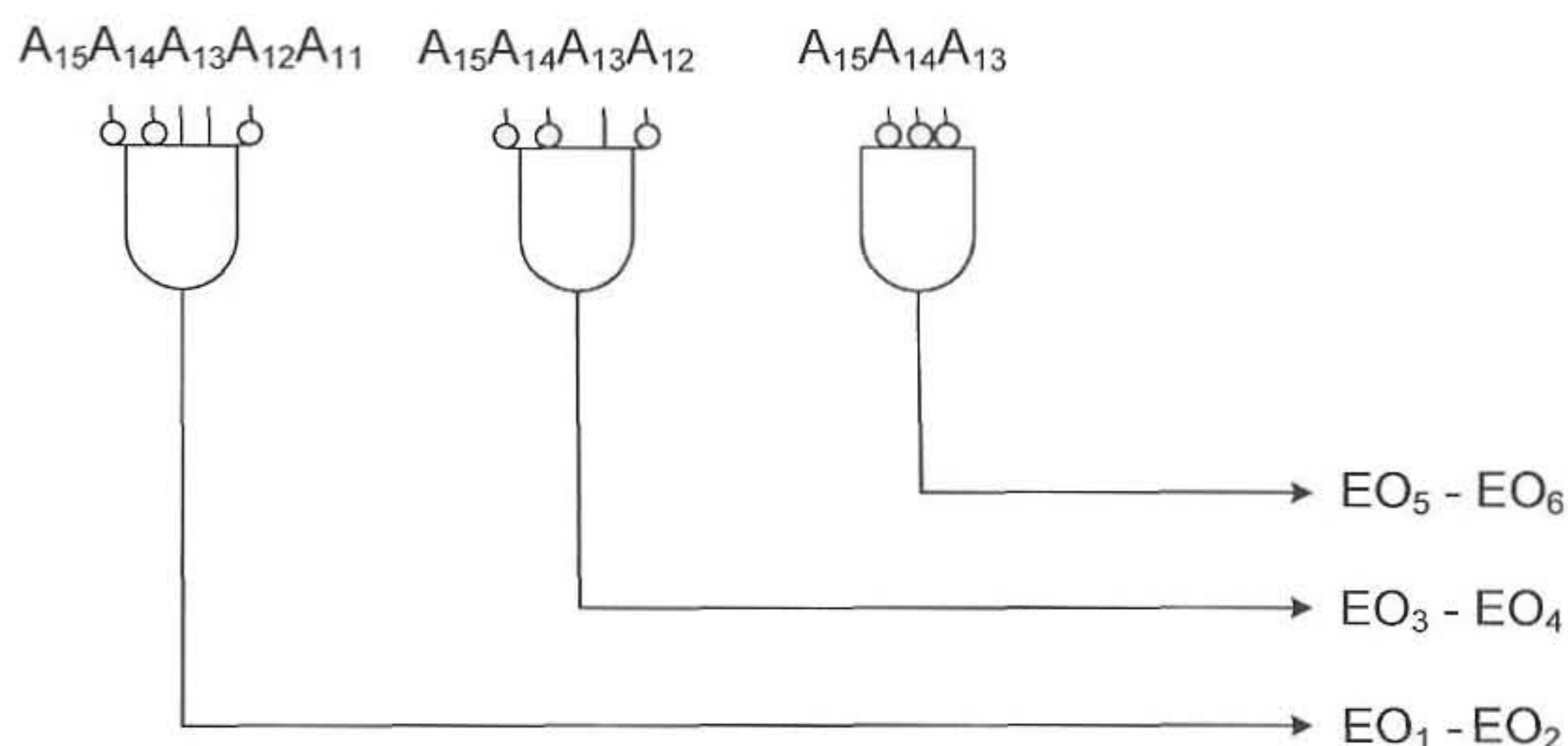
Πίνακας 5.2.4

Χάρτης διευθύνσεων του συστήματος μνήμης.

ολοκληρωμένο κύκλωμα μνήμης	πλήθος εισόδων διεύθυνσης	Διευθύνσεις		
		A_{15}	...	$A_1 A_0$
SRAM5 - SRAM6	13	από	0000000000000000	
		έως	0001111111111111	
SRAM3 - SRAM4	12	από	0010000000000000	
		έως	0010111111111111	
SRAM1 - SRAM2	11	από	0011000000000000	
		έως	0011011111111111	



Σχήμα 5.2.5 Μονάδα ημιαγωγικής μνήμης 28 Kψηφιολέξεων με δύο ψηφιολέξεις ανά θέση μνήμης.



Σχήμα 5.2.6 Παραγωγή των σημάτων ενεργοποίησης με πύλες AND.

Λόγω των OK μνήμης που έχουμε στη διάθεσή μας δεν είναι δυνατόν να σχεδιάσουμε σύστημα μνήμης με περισσότερες από δύο ψηφιολέξεις ανά θέση μνήμης, χωρίς να αφήσουμε κάποια OK αχρησιμοποίητα.

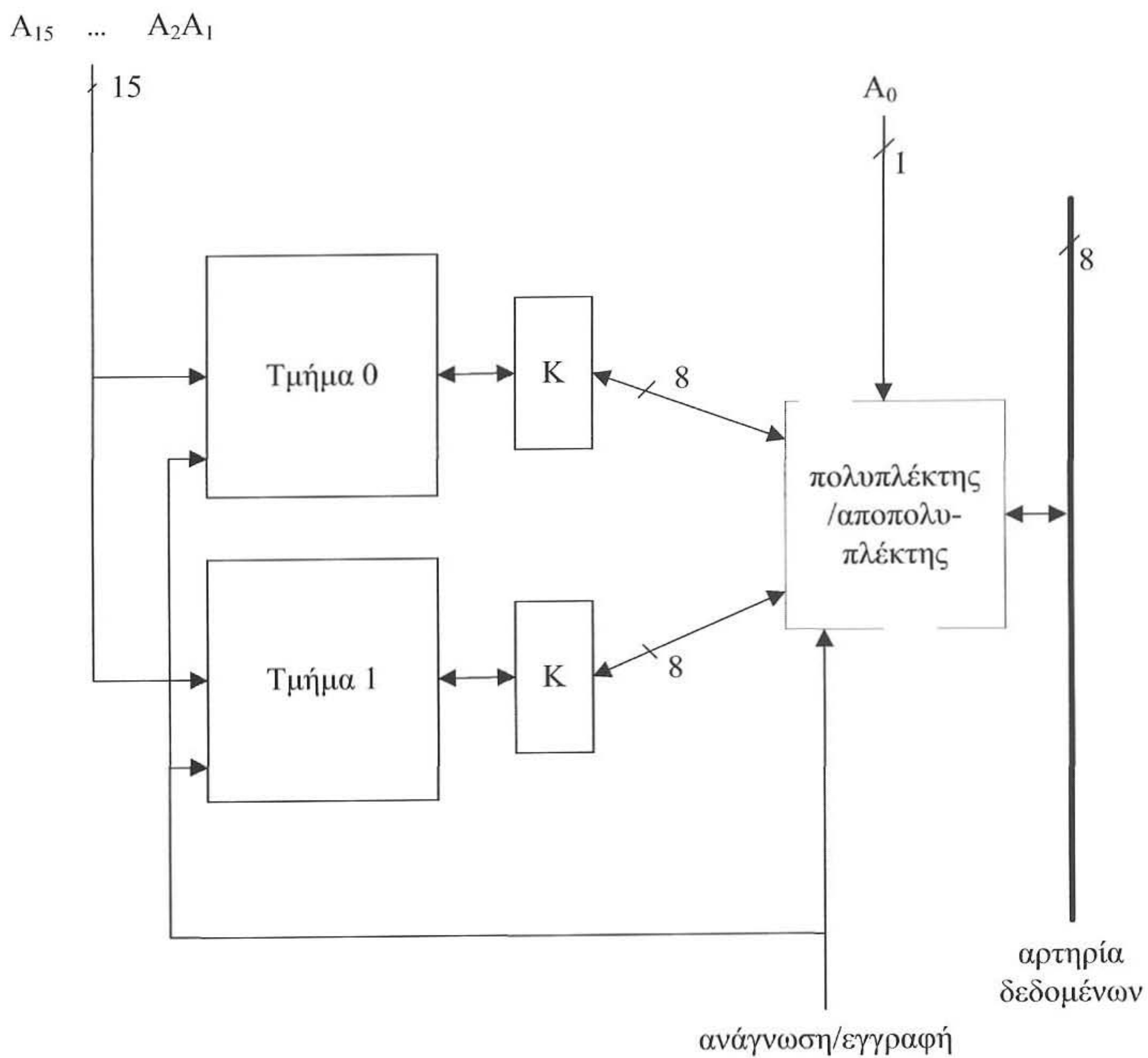
B. Σχεδίαση συστήματος μνήμης 2 τρόπων χαμηλής τάξης διαφύλωσης.

B.1 Σε κάθε μια ψηφιολέξη του συστήματος μνήμης αντιστοιχεί μια διεύθυνση.

Αφού έχουμε σύστημα μνήμης 2 τρόπων χαμηλής τάξης διαφύλωσης θα έχουμε 2 τμήματα. Επομένως το λιγότερο σημαντικό δυαδικό ψηφίο A_0 της διεύθυνσης χρησιμοποιείται για την επιλογή του τμήματος ενώ τα υπόλοιπα δυαδικά ψηφία της διεύθυνσης διευθυνσιοδοτούν παράλληλα και τα δύο τμήματα μνήμης (βλέπε σχήμα 5.2.7). Τα δύο τμήματα μνήμης πρέπει να έχουν το ίδιο πλήθος θέσεων μνήμης.

Στους Πίνακες 5.2.5 και 5.2.6 τοποθετήσαμε τα OK στο χώρο διευθύνσεων του συστήματος μνήμης έτσι ώστε η διεύθυνση $A_{15} \dots A_2 A_1$ που αντιστοιχεί στην πρώτη θέση κάθε OK μνήμης να είναι ακέραιο πολλαπλάσιο της χωρητικότητας του συγκεκριμένου OK. Στα σχήματα 5.2.8 και 5.2.9 δίνονται τα τμήματα 0 και 1 της μνήμης που αντιστοιχούν στους χάρτες διευθύνσεων των Πινάκων 5.2.5 και 5.2.6 αντίστοιχα. Και πάλι αντί της χρησιμοποίησης των αποκωδικοποιητών 3 σε 8 θα μπορούσαμε να παράγουμε τα σήματα ενεργοποίησης των OK μνήμης χρησιμοποιώντας πύλες AND.

Στο σχήμα 5.2.10 δίνεται το σύστημα μνήμης χαμηλής τάξης διαφύλωσης με δύο τμήματα. Επειδή οι αποκωδικοποιητές των τμημάτων 0 και 1 (βλ. σχήματα 5.2.8 και 5.2.9) είναι ίδιοι και λαμβάνουν τις ίδιες εισόδους, στο σχήμα 5.2.10 αντικαταστάθηκαν από ένα αποκωδικοποιητή.



Σχήμα 5.2.7 Σύστημα μνήμης χαμηλής τάξης διαφύλωσης με δύο τμήματα.

Πίνακας 5.2.5

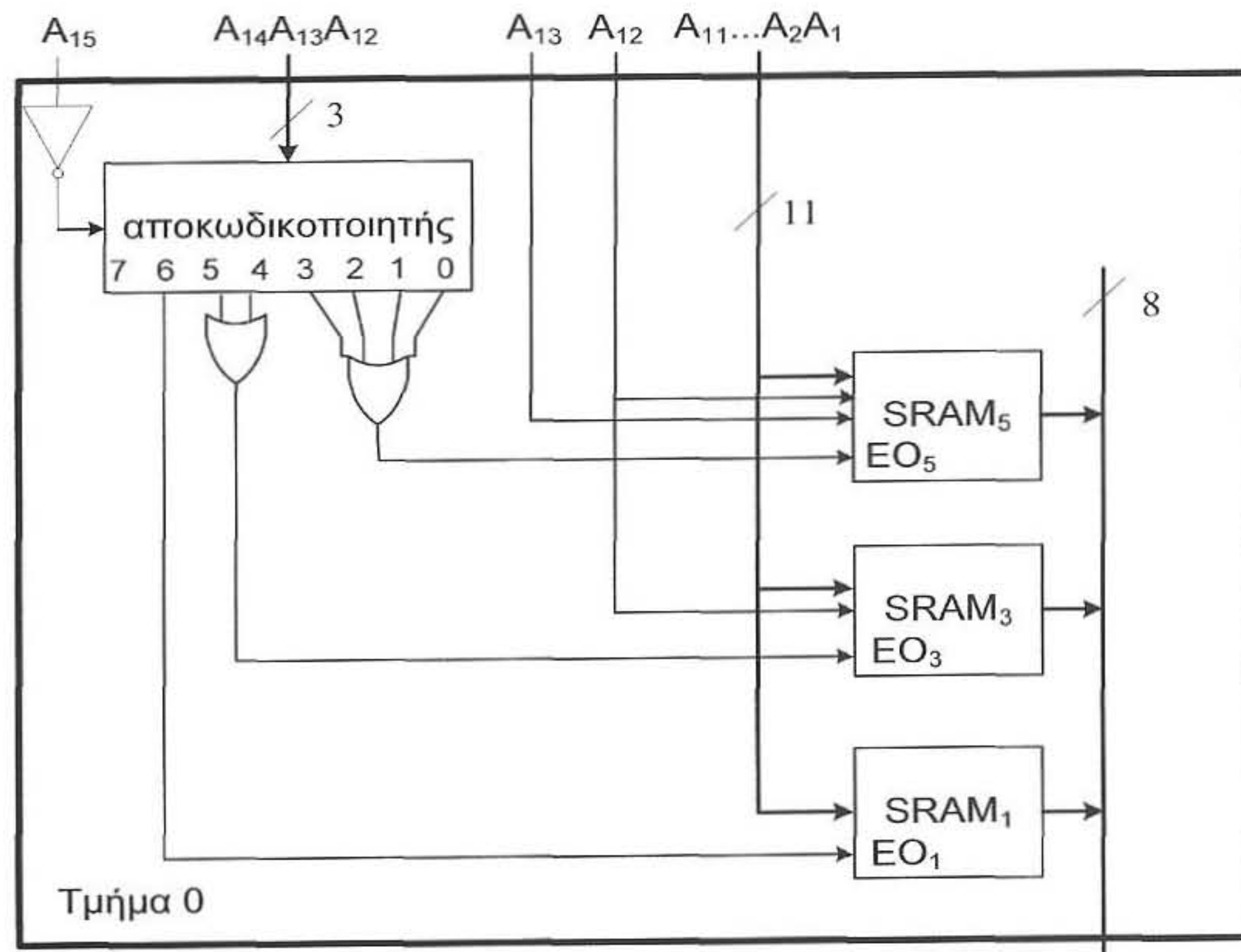
Χάρτης διευθύνσεων του τμήματος 0 ($A_0=0$) του συστήματος μνήμης.

ολοκληρωμένο κύκλωμα μνήμης	πλήθος εισόδων διεύθυνσης	Διευθύνσεις		
			A_{15}	A_1
SRAM5	13	από	0000000000000000	
		έως	0011111111111111	
SRAM3	12	από	0100000000000000	
		έως	0101111111111111	
SRAM1	11	από	0110000000000000	
		έως	0110111111111111	

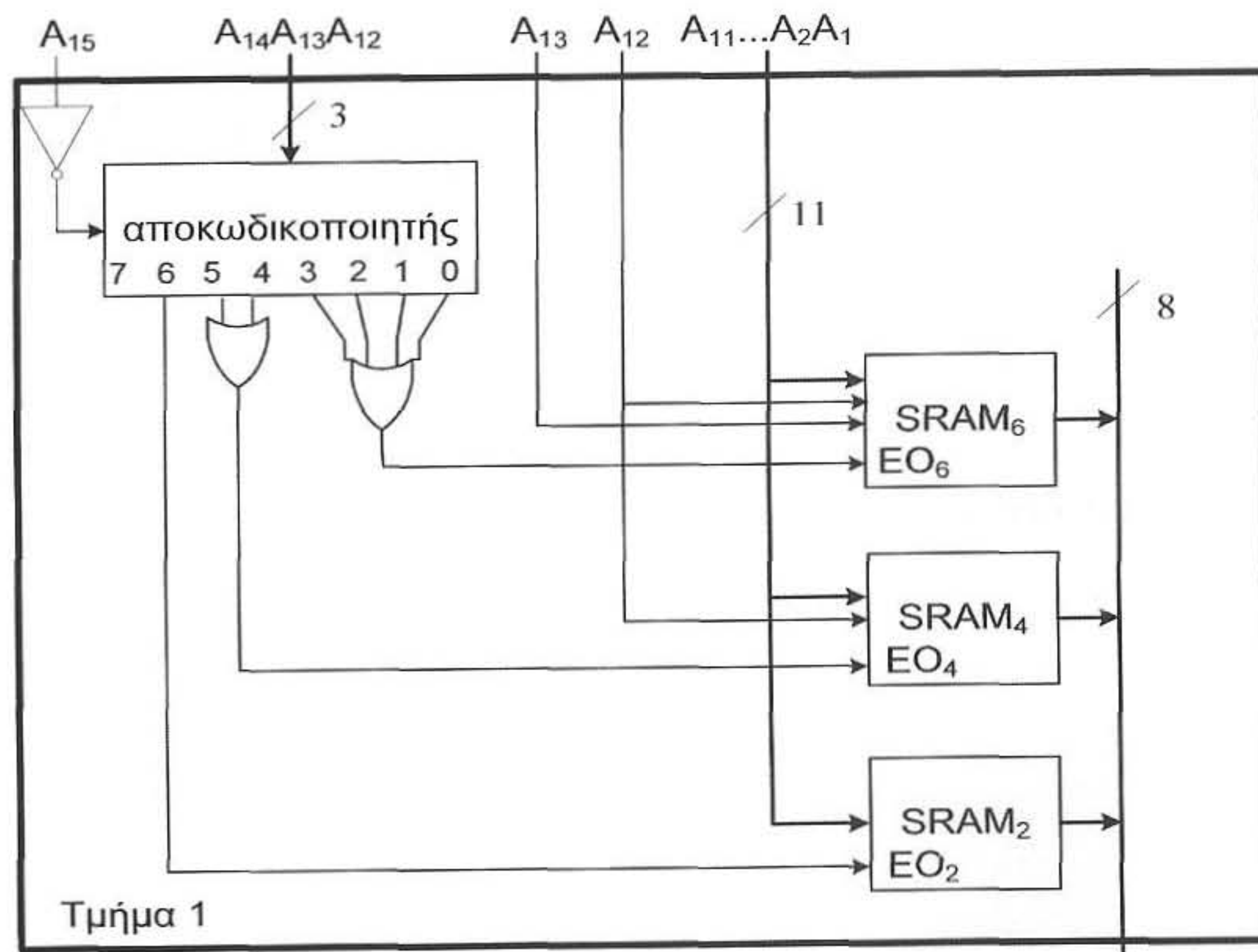
Πίνακας 5.2.6

Χάρτης διευθύνσεων του τμήματος 1 ($A_0=1$) του συστήματος μνήμης.

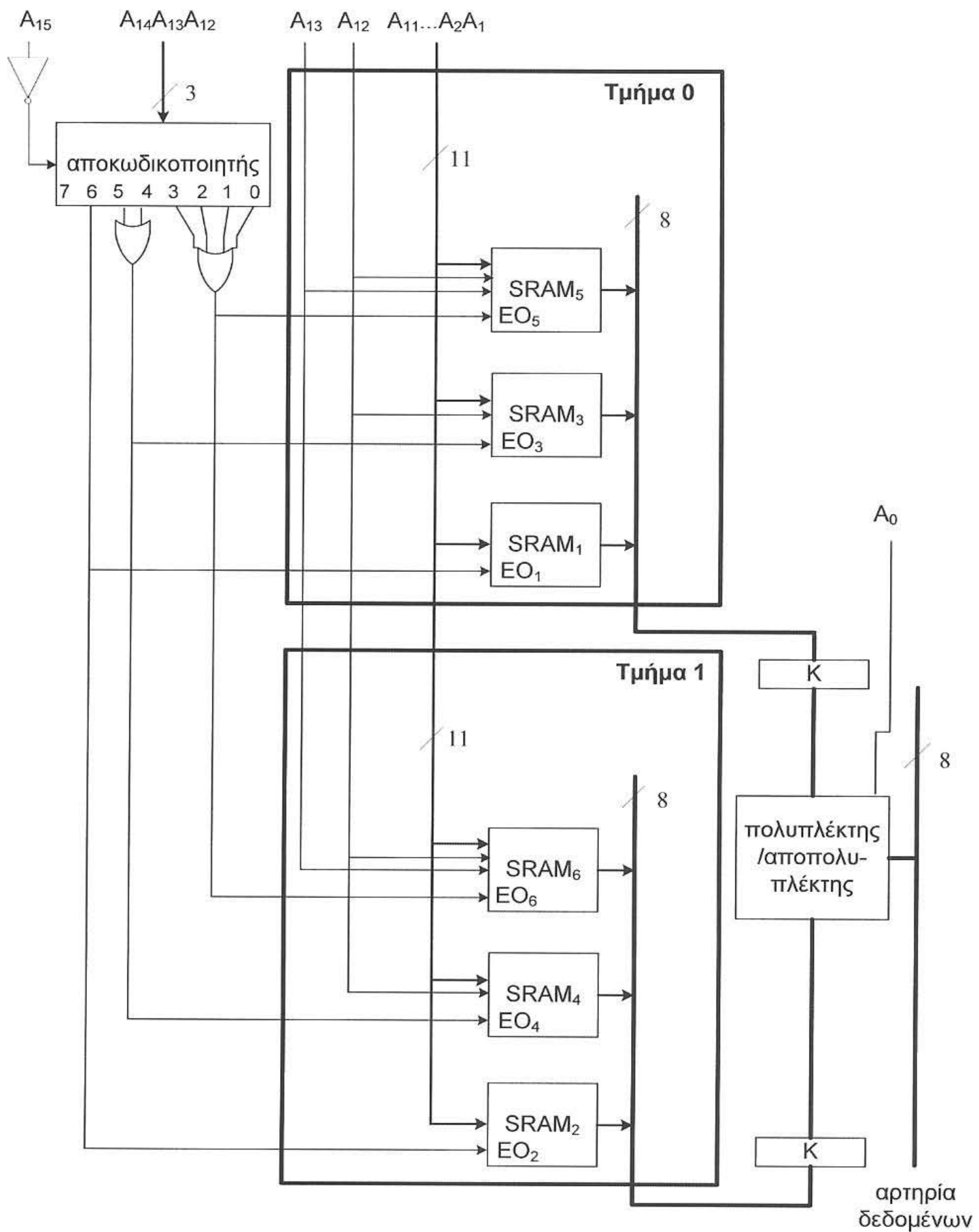
ολοκληρωμένο κύκλωμα μνήμης	πλήθος εισόδων διεύθυνσης	Διευθύνσεις		
			A_{15}	A_1
SRAM6	13	από	0000000000000000	
		έως	0011111111111111	
SRAM4	12	από	0100000000000000	
		έως	0101111111111111	
SRAM2	11	από	0110000000000000	
		έως	0110111111111111	



Σχήμα 5.2.8 Δομή Τμήματος 0.



Σχήμα 5.2.9 Δομή Τμήματος 1.



Σχήμα 5.2.10 Το σύστημα μνήμης χαμηλής τάξης διαφύλλωσης με δύο τμήματα.

Λόγω των OK μνήμης που έχουμε στη διάθεσή μας και του γεγονότος ότι τα τμήματα της μνήμης πρέπει να έχουν το ίδιο πλήθος θέσεων μνήμης, δεν είναι δυνατόν να σχεδιάσουμε σύστημα μνήμης με περισσότερα από δύο τμήματα χωρίς να αφήσουμε κάποια OK αχρησιμοποίητα.

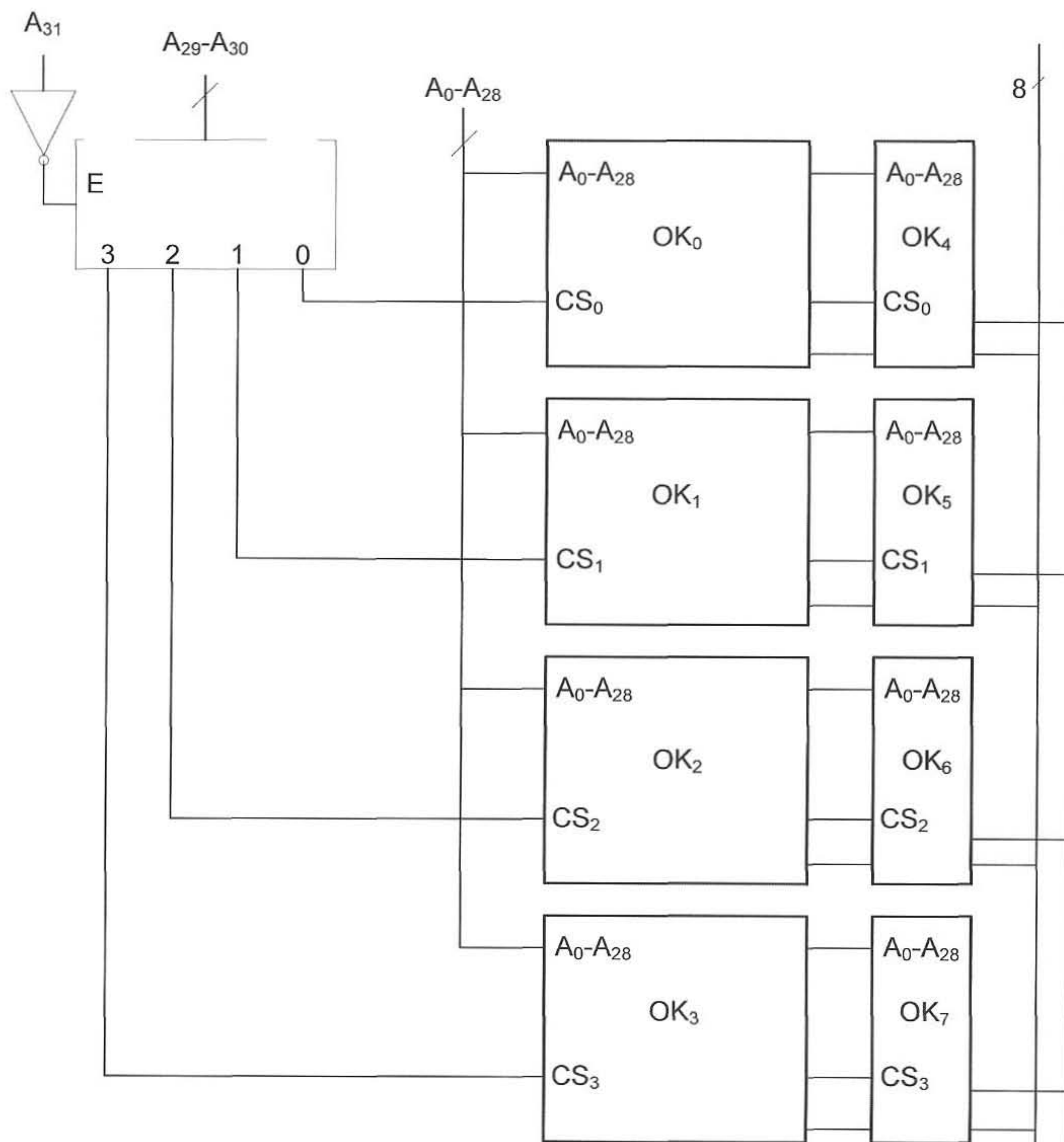
Άσκηση 5.3

Σ' ένα υπολογιστή ο επεξεργαστής έχει αρτηρία διευθύνσεων των 32 δυαδικών ψηφίων. Έχετε στη διάθεση σας ολοκληρωμένα κυκλώματα, OK, μνήμης με χωρητικότητα 4096 M δυαδικών ψηφίων (M bit) και οργάνωση μιας ψηφιολέξης ανά θέση μνήμης και OK μνήμης με χωρητικότητα 512 M δυαδικών ψηφίων και οργάνωση ενός δυαδικού ψηφίου ανά θέση μνήμης. Να σχεδιάσετε σύστημα μνήμης με χωρητικότητα 18.432 M δυαδικών ψηφίων με εννέα δυαδικά ψηφία ανά θέση μνήμης, εκ των οποίων 8 δυαδικά ψηφία δεδομένων και 1 δυαδικό ψηφίο ισοτιμίας.

Απάντηση

Αφού τα OK μνήμης με χωρητικότητα 4096 M δυαδικών ψηφίων (M bit) έχουν οργάνωση μιας ψηφιολέξης ανά θέση μνήμης, συνεπάγεται ότι έχουν $(4096M)/8=512$ M θέσεις μνήμης το κάθε ένα. Άρα κάθε ένα OK μνήμης θα έχει 29 εισόδους διεύθυνσης.

Αφού θέλουμε να σχεδιάσουμε σύστημα μνήμης με χωρητικότητα 18.432 M δυαδικών ψηφίων με εννέα δυαδικά ψηφία ανά θέση μνήμης, συνεπάγεται ότι το σύστημα μνήμης που θέλουμε να σχεδιάσουμε θα έχει $(18.432 M)/9 = 2048$ M θέσεις των 9 δυαδικών ψηφίων η κάθε μια. Συνδυάζοντας ένα OK με χωρητικότητα 4096 M δυαδικών ψηφίων (512 M θέσεις μνήμης των 8 δυαδικών ψηφίων η κάθε μια) και ένα OK με χωρητικότητα 512 M δυαδικών ψηφίων (512 M θέσεις του ενός δυαδικού ψηφίου) παίρνουμε μία μνήμη με 512 M θέσεις των 9 δυαδικών ψηφίων η κάθε μία. Αφού θέλουμε το σύστημα μνήμης να έχει συνολικά 2048 M θέσεις των 9 δυαδικών ψηφίων θα χρησιμοποιήσουμε $(2048 M)/(512 M)=4$ OK μνήμης με οργάνωση των 8 δυαδικών ψηφίων ανά θέση και 4 OK μνήμης με οργάνωση 1 δυαδικού ψηφίου ανά θέση. Στο σχήμα 5.3.1 δίνεται το μπλοκ διάγραμμα του ζητούμενου συστήματος μνήμης.



Σχήμα 5.3.1

Άσκηση 5.4

Σ' ένα υπολογιστή η αρτηρία διευθύνσεων είναι των 32 δυαδικών ψηφίων. Έχετε στη διάθεσή σας ολοκληρωμένα κυκλώματα των 256 Μψηφιολέξεων το κάθε ένα με οργάνωση μίας ψηφιολέξης ανά θέση μνήμης και αποκωδικοποιητές με είσοδο ενεργοποίησης. Να σχεδιάσετε σύστημα κύριας μνήμης με οργάνωση μίας ψηφιολέξης ανά θέση μνήμης έτσι ώστε η κύρια μνήμη να είναι επεκτάσιμη κατά 1 Γψηφιολέξεις μέχρι το μέγιστο της χωρητικότητάς της. Θεωρήστε ότι το εύρος της αρτηρίας δεδομένων είναι 8. Να δώσετε τη δομή του συστήματος μνήμης ώστε να μπορεί να γίνεται η επέκταση μέχρι τη μέγιστη χωρητικότητα καθώς και τη δομή κάθε τμήματος με χωρητικότητα 1 G ψηφιολέξεις, το οποίο χρησιμοποιείται κάθε φορά για την επέκταση.

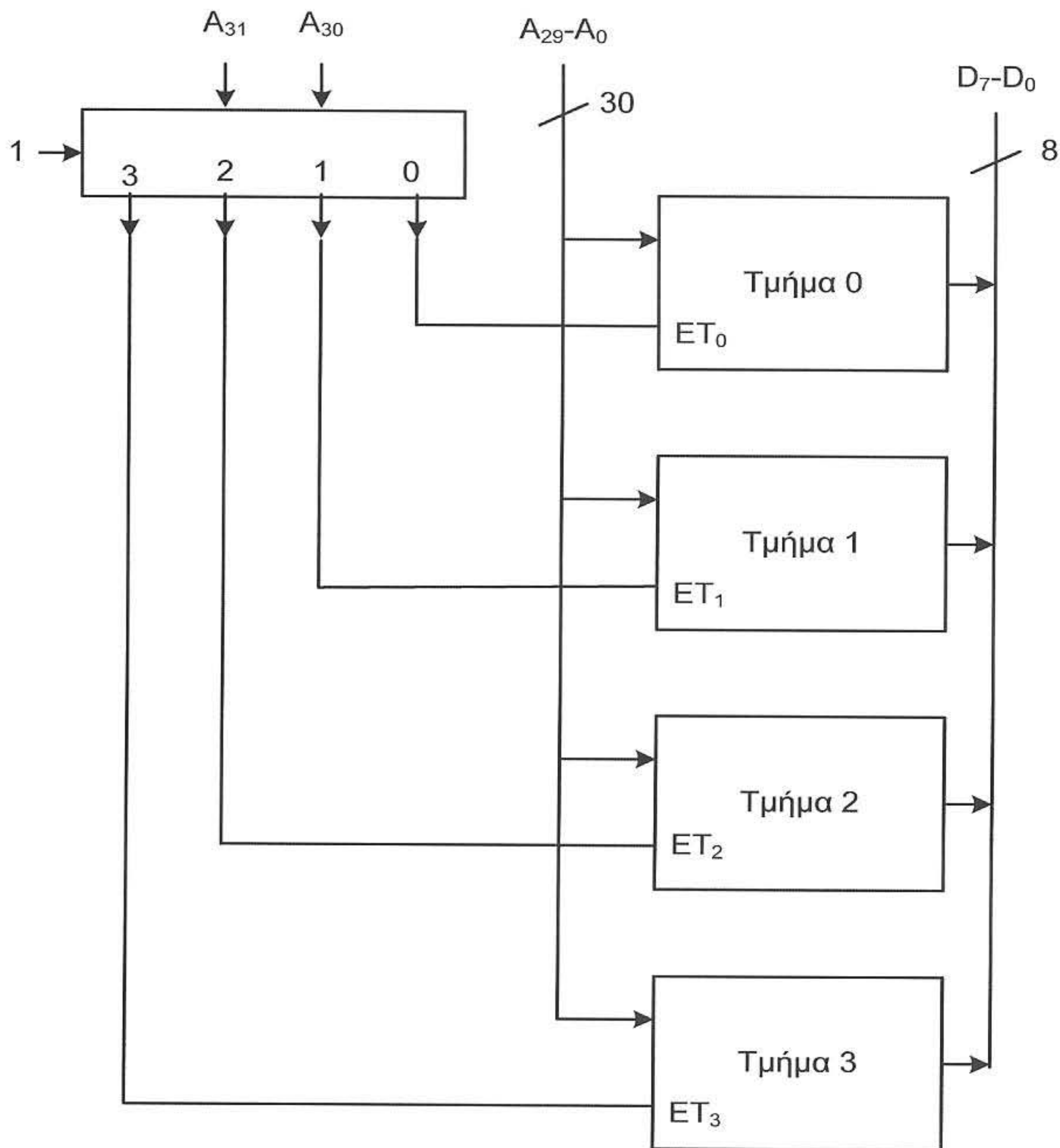
Απάντηση

Εφόσον η αρτηρία διευθύνσεων είναι των 32 δυαδικών ψηφίων μπορούμε να διευθυνσιοδοτήσουμε έως 2^{32} θέσεις μνήμης. Θέλουμε να σχεδιάσουμε κύρια μνήμη επεκτάσιμη κατά 1 Γψηφιολέξεις κάθε φορά. Επομένως η κύρια μνήμη θα αποτελείται από ένα έως τέσσερα τμήματα μνήμης που κάθε τμήμα μνήμης θα έχει χωρητικότητα 1 Γψηφιολέξεις (βλ. Πίνακα 5.4.1). Για την διευθυνσιοδότηση ενός τμήματος μνήμης με χωρητικότητα 1 Γψηφιολέξεων απαιτούνται τα 30 λιγότερο σημαντικά δυαδικά ψηφία της διεύθυνσης, δηλαδή τα δυαδικά ψηφία $A_{29} \dots A_1 A_0$. Τα υπόλοιπα δύο πιο σημαντικά δυαδικά ψηφία $A_{31} A_{30}$ της διεύθυνσης θα οδηγούν έναν αποκωδικοποιητή 2 σε 4. Κάθε έξοδος του αποκωδικοποιητή θα επιλέγει και ένα τμήμα του συστήματος μνήμης. Η δομή του επεκτάσιμου συστήματος μνήμης δίνεται στο σχήμα 5.4.1. Η είσοδος ενεργοποίησης του αποκωδικοποιητή 2 σε 4 οδηγείται μόνιμα με τη λογική τιμή 1.

Πίνακας 5.4.1

Χάρτης διευθύνσεων του συστήματος μνήμης.

Τμήμα	πλήθος εισόδων διεύθυνσης	Διευθύνσεις				
		$A_{31}A_{30}$	$A_{29}A_{28}$...	A_1A_0	
T_0	30	από	0 0	0 0	...	0 0
		έως	0 0	1 1	...	1 1
T_1	30	από	0 1	0 0	...	0 0
		έως	0 1	1 1	...	1 1
T_2	30	από	1 0	0 0	...	0 0
		έως	1 0	1 1	...	1 1
T_3	30	από	1 1	0 0	...	0 0
		έως	1 1	1 1	...	1 1



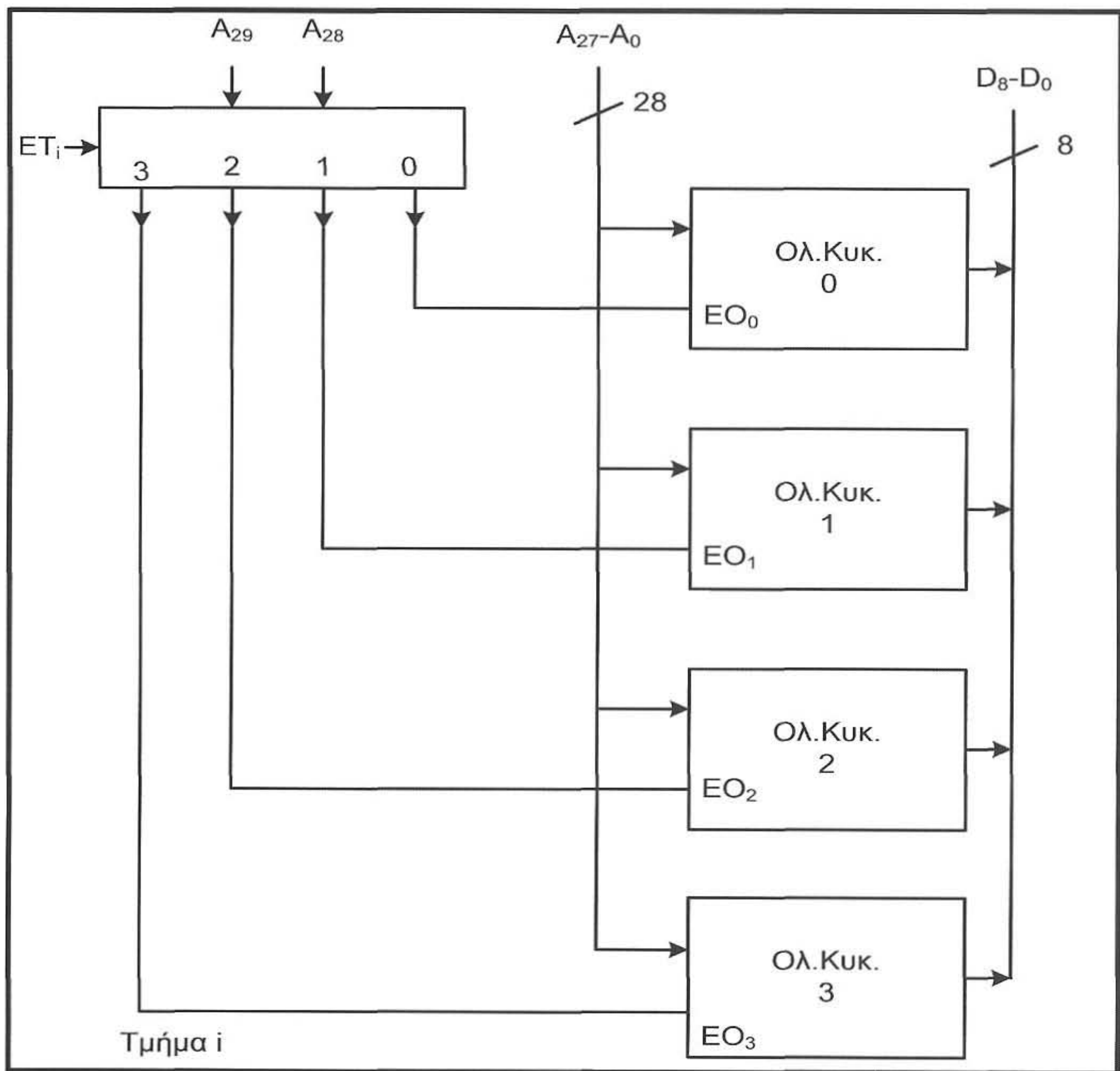
Σχήμα 5.4.1 Η δομή του επεκτάσιμου συστήματος μνήμης.

Για τη σχεδίαση καθενός τμήματος μνήμης με χωρητικότητα 1 Gψηφιολέξεων θα χρησιμοποιηθούν 4 ολοκληρωμένα κυκλώματα μνήμης με χωρητικότητα 256 M ψηφιολέξεων το καθένα (βλ. Πίνακα 5.4.2). Από τα 30 λιγότερο σημαντικά δυαδικά ψηφία $A_{29} \dots A_1 A_0$ της διεύθυνσης, τα οποία χρησιμοποιούνται για την διευθυνσιοδότηση κάθε τμήματος, τα 28 λιγότερο σημαντικά δυαδικά ψηφία $A_{27} \dots A_1 A_0$ θα χρησιμοποιηθούν για την διευθυνσιοδότηση κάθε ολοκληρωμένου κυκλώματος μνήμης. Τα υπόλοιπα δύο δυαδικά ψηφία $A_{29} A_{28}$ οδηγούν έναν αποκωδικοποιητή 2 σε 4.

Πίνακας 5.4.2
Χάρτης διευθύνσεων ενός τμήματος.

Ολοκληρωμένο κύκλωμα μνήμης	πλήθος εισόδων διεύθυνσης		Διευθύνσεις				
			$A_{29}A_{28}$	$A_{27}A_{26}$...	A_1A_0	
OK_0	28	από	0 0	0 0	...	0 0	
		έως	0 0	1 1	...	1 1	
OK_1	28	από	0 1	0 0	...	0 0	
		έως	0 1	1 1	...	1 1	
OK_2	28	από	1 0	0 0	...	0 0	
		έως	1 0	1 1	...	1 1	
OK_3	28	από	1 1	0 0	...	0 0	
		έως	1 1	1 1	...	1 1	

Καθεμία από τις εξόδους του αποκωδικοποιητή οδηγεί την είσοδο ενεργοποίησης ολοκληρωμένου ΕΟ ενός από τα τέσσερα ολοκληρωμένα κυκλώματα μνήμης. Η δομή ενός τμήματος δίνεται στο σχήμα 5.4.2. Η είσοδος ενεργοποίησης του αποκωδικοποιητή 2 σε 4 του σχήματος 5.4.2 οδηγείται από την αντίστοιχη έξοδο του αποκωδικοποιητή 2 σε 4 με εισόδους A_{30} και A_{31} του σχήματος 5.4.1.



Σχήμα 5.4.2 Δομή ενός τμήματος. Ένα τέτοιο τμήμα μπορεί να συνδεθεί στη θέση ενός τμήματος του σχήματος 5.4.1.

Άσκηση 5.5 (5.2 βιβλίου)

Ένας μαγνητικός δίσκος με $\tau = 400.000$ ομόκεντρους κύκλους (tracks) συνολικά σε 4 επιφάνειες εγγραφής, περιστρέφεται με ταχύτητα 10000 στροφές το λεπτό (rpm) και ο χρόνος μετακίνησης της κεφαλής από ένα ομόκεντρο κύκλο στον γειτονικό του είναι 0,7 ms. Ποιος είναι ο ελάχιστος χρόνος που απαιτείται για να κάνουμε ένα πλήρες αντίγραφο (backup) του περιεχομένου του μαγνητικού δίσκου; Υποθέστε ότι ο δίσκος είναι γεμάτος, έχουμε μία κεφαλή εγγραφής/ανάγνωσης ανά επιφάνεια εκ των οποίων μόνο μία διαβάζει ή γράφει κάθε φορά και ότι η συσκευή στην οποία παίρνουμε το αντίγραφο μπορεί να δέχεται δεδομένα με το ρυθμό που τα προσφέρει ο μαγνητικός δίσκος.

Απάντηση

Αφού η άσκηση μας ζητάει να βρούμε τον ελάχιστο χρόνο για να κάνουμε ένα αντίγραφο, θα θεωρήσουμε ότι η αντιγραφή γίνεται ανά κύλινδρο, δηλαδή πρώτα αντιγράφονται τα περιεχόμενα όλων των ομόκεντρων κύκλων με την ίδια ακτίνα και μετά τα περιεχόμενα του γειτονικού κυλίνδρου κλπ ξεκινώντας από τον εξωτερικό κύλινδρο. Θεωρούμε ότι οι κεφαλές κατά την έναρξη βρίσκονται στον εξωτερικό κύλινδρο. Επομένως το πλήθος των μετακινήσεων από κύλινδρο σε κύλινδρο ισούται με $(\tau/4)-1=(400.000/4)-1=99.999$. Ο απαιτούμενος χρόνος για μία πλήρη περιστροφή το δίσκου ισούται με $T_{\text{περ}}=(1/10000)$ λεπτά $= (60000/10000)$ ms $= 6$ ms. Αφού θέλουμε τον ελάχιστο χρόνο που απαιτείται για να κάνουμε ένα αντίγραφο, θα θεωρήσουμε ότι κατά την έναρξη η κεφαλή βρίσκεται στη σωστή θέση του εξωτερικού κυλίνδρου για να αρχίσει η ανάγνωση. Επομένως ο απαιτούμενος συνολικός χρόνος είναι:

$$\begin{aligned} T &= 4 \times T_{\text{περ}} + 99.999 \times (0,7 \text{ ms} + 4 \times T_{\text{περ}}) = \\ &= 4 \times 6 \text{ ms} + 99.999 \times (0,7 \text{ ms} + 4 \times 6 \text{ ms}) = \\ &= 2.469.999,3 \text{ ms} \approx 2.469,9993 \text{ s} \approx 41,17 \text{ λεπτά.} \end{aligned}$$

Άσκηση 5.6 (5.3 βιβλίου)

α. Να υπολογίσετε το μέσο χρόνο προσπέλασης του ιεραρχικού συστήματος μνήμης που χαρακτηρίζεται από τα στοιχεία του Πίνακα 5.6.1

β. Υποθέστε ότι για λόγους ελάττωσης του κόστους πρέπει να αφαιρεθεί ένα από τα επίπεδα κρυφής μνήμης. Ποιο πρέπει να αφαιρεθεί ώστε να επηρεαστεί λιγότερο ο μέσος χρόνος προσπέλασης του ιεραρχικού συστήματος μνήμης;

Πίνακας 5.6.1

Μνήμη	Μέσος χρόνος προσπέλασης seconds	Λόγος επιτυχίας
κρυφή μνήμη 1 ^{ου} επιπέδου	2×10^{-9}	0.990000
κρυφή μνήμη 2 ^{ου} επιπέδου	20×10^{-9}	0.999900
κύρια μνήμη	400×10^{-9}	0.999990
βοηθητική	10×10^{-3}	1.000000

Απάντηση

α. Ο μέσος χρόνος προσπέλασης T δίνεται από την σχέση:

$$T = \sum_{i=1}^4 (E_i - E_{i-1}) \times T_i \quad \text{όπου } E_i \text{ είναι ο λόγος επιτυχίας του επιπέδου } i \text{ του συστήματος}$$

μνήμης. Τότε $E_0=0$ (διότι δεν υπάρχει επίπεδο μνήμης 0) και $E_4=1$ (διότι στη βοηθητική μνήμη υπάρχει όλη η πληροφορία).

$$\begin{aligned} \text{Επομένως } T &= 0,99 \times 2 \times 10^{-9} + (0,9999 - 0,99) \times 20 \times 10^{-9} + (0,99999 - 0,9999) \times 400 \times 10^{-9} + \\ &+ (1 - 0,99999) \times 10 \times 10^{-3} \text{ s} = \\ &= 0,99 \times 2 \times 10^{-9} + 0,0099 \times 20 \times 10^{-9} + 0,00009 \times 400 \times 10^{-9} + 0,00001 \times 10^7 \times 10^{-9} \text{ s} = \\ &= (0,99 \times 2 + 0,0099 \times 20 + 0,00009 \times 400 + 0,00001 \times 10^7) \times 10^{-9} \text{ s} = \\ &= 102,214 \times 10^{-9} \text{ s} = 102,214 \text{ ns}. \end{aligned}$$

β. Θα εξετάσουμε κάθε μια περίπτωση ξεχωριστά. Εάν αφαιρέσουμε την κρυφή μνήμη 1^{ου} επιπέδου, τότε:

$$\begin{aligned} T_1 &= 0,9999 \times 20 \times 10^{-9} + (0,99999 - 0,9999) \times 400 \times 10^{-9} + (1 - 0,99999) \times 10 \times 10^{-3} \text{ s} = \\ &= 0,9999 \times 20 \times 10^{-9} + 0,00009 \times 400 \times 10^{-9} + 0,00001 \times 10^7 \times 10^{-9} \text{ s} = \\ &= (0,9999 \times 20 + 0,00009 \times 400 + 0,00001 \times 10^7) \times 10^{-9} \text{ s} = 120,034 \times 10^{-9} \text{ s} = 120,034 \text{ ns} \end{aligned}$$

Εάν αφαιρέσουμε την κρυφή μνήμη 2^{ου} επιπέδου, τότε:

$$\begin{aligned} T_2 &= 0,99 \times 2 \times 10^{-9} + (0,99999 - 0,99) \times 400 \times 10^{-9} + (1 - 0,99999) \times 10 \times 10^{-3} \text{ s} = \\ &= 0,99 \times 2 \times 10^{-9} + 0,00999 \times 400 \times 10^{-9} + 0,00001 \times 10^7 \times 10^{-9} \text{ s} = \\ &= (0,99 \times 2 + 0,00999 \times 400 + 0,00001 \times 10^7) \times 10^{-9} \text{ s} = 105,976 \times 10^{-9} \text{ s} = 105,976 \text{ ns} \end{aligned}$$

Από τα ανωτέρω, συνεπάγεται ότι πρέπει να αφαιρεθεί η κρυφή μνήμη 2^{ου} επιπέδου.

Άσκηση 5.7 (5.4 βιβλίου)

Ένας σχεδιαστής πρέπει να επιλέξει μεταξύ των ακόλουθων δύο επιλογών.

α. Υλοποίηση κρυφής μνήμης οργάνωσης μονοσήμαντης απεικόνισης με λόγο επιτυχίας $h_1=0,85$ και χρόνο προσπέλασης $t_1=9$ ns.

β. Υλοποίηση κρυφής μνήμης 4-τρόπων συνόλου συσχέτισης με ρυθμό επιτυχίας $h_2=0,95$ και χρόνο προσπέλασης $t_2=12$ ns.

Ποια υλοποίηση πρέπει να επιλέξει ο σχεδιαστής ώστε να επιτύχει τον ελάχιστο μέσο χρόνο προσπέλασης του συστήματος μνήμης; Να θεωρήσετε ότι η ΚΜΕ προσπελαύνει πληροφορία μόνο από την κρυφή μνήμη.

Απάντηση

Έστω ότι $t_{\text{μπλοκ}}$ είναι ο χρόνος μεταφοράς ενός μπλοκ πληροφορίας από τη κύρια μνήμη στην κρυφή μνήμη. Ο μέσος χρόνος προσπέλασης T' και T'' στην περίπτωση της κρυφής μνήμης οργάνωσης μονοσήμαντης απεικόνισης και 4-τρόπων συνόλου συσχέτισης αντίστοιχα είναι:

$$T' = h_1 \times t_1 + (1 - h_1) \times (t_1 + t_{\text{μπλοκ}})$$

$$T'' = h_2 \times t_2 + (1 - h_2) \times (t_2 + t_{\text{μπλοκ}})$$

Η πρώτη λύση θα είναι καλύτερη όταν $T'' - T' > 0$ ή ισοδύναμα

$$h_2 \times t_2 + (1 - h_2) \times (t_2 + t_{\text{μπλοκ}}) - h_1 \times t_1 - (1 - h_1) \times (t_1 + t_{\text{μπλοκ}}) > 0 \quad \text{ή}$$

$$h_2 \times t_2 + (1 - h_2) \times t_2 + (1 - h_2) \times t_{\text{μπλοκ}} - h_1 \times t_1 - (1 - h_1) \times t_1 - (1 - h_1) \times t_{\text{μπλοκ}} > 0 \quad \text{ή}$$

$$h_2 \times t_2 + (1 - h_2) \times t_2 - h_1 \times t_1 - (1 - h_1) \times t_1 + (1 - h_2) \times t_{\text{μπλοκ}} - (1 - h_1) \times t_{\text{μπλοκ}} > 0 \quad \text{ή}$$

$$h_2 \times t_2 + t_2 - h_2 \times t_2 - h_1 \times t_1 - t_1 + h_1 \times t_1 + [(1 - h_2) - (1 - h_1)] \times t_{\text{μπλοκ}} > 0 \quad \text{ή}$$

$$t_2 - t_1 + (h_1 - h_2) \times t_{\text{μπλοκ}} > 0 \quad \text{ή}$$

$$t_2 - t_1 > - (h_1 - h_2) \times t_{\text{μπλοκ}} \quad \text{ή}$$

$$t_2 - t_1 > (h_2 - h_1) \times t_{\text{μπλοκ}} \quad \text{ή}$$

$$(t_2 - t_1) / (h_2 - h_1) > t_{\text{μπλοκ}} \quad \text{ή}$$

$$= (12 - 9) \text{ ns} / (0,95 - 0,85) = 3 \text{ ns} / 0,1 = 30 \text{ ns} > t_{\text{μπλοκ}}$$

Επομένως η πρώτη λύση είναι καλύτερη όταν η τιμή του $t_{\text{μπλοκ}}$ είναι μικρότερη των 30 ns.

Άσκηση 5.8 (5.5 βιβλίου)

Θεωρήστε ένα υπολογιστή με εύρος αρτηρίας διευθύνσεων των 32 δυαδικών ψηφίων και ότι η κύρια μνήμη είναι οργανωμένη σε λέξεις των 16 δυαδικών ψηφίων ανά θέση μνήμης. Θεωρήστε επίσης ότι η κρυφή μνήμη έχει οργάνωση 4-τρόπων συνόλου συσχέτισης με χωρητικότητα 64 Κλέξεις και τέσσερις λέξεις των 16 δυαδικών ψηφίων ανά πλαίσιο.

α. Πόσα μπλοκ της κύριας μνήμης μπορούν να αποθηκευτούν στο ίδιο σύνολο της κρυφής μνήμης (προφανώς όχι ταυτόχρονα);

β. Προσδιορίστε το σύνολο της κρυφής μνήμης στο οποίο μπορεί να αποθηκευτεί η λέξη που είναι αποθηκευμένη στη θέση με διεύθυνση $960F8_{(16)}$.

γ. Προσδιορίστε τις διευθύνσεις όλων των μπλοκ της κύριας μνήμης που μπορούν να αποθηκευτούν στο σύνολο με διεύθυνση $60_{(16)}$.

Απάντηση

α. Η κρυφή μνήμη είναι των 64 Κλέξεων, δηλαδή 2^{16} λέξεις και κάθε πλαίσιο αποτελείται από 4 λέξεις, επομένως η κρυφή μνήμη περιέχει $2^{16}/2^2 = 2^{16-2} = 2^{14}$ πλαίσια. Κάθε σύνολο αποτελείται από 4 πλαίσια, επομένως η κρυφή μνήμη περιέχει $2^{14}/2^2 = 2^{14-2} = 2^{12}$ σύνολα. Εφόσον η αρτηρία διευθύνσεων είναι των 32 δυαδικών ψηφίων, η κύρια μνήμη μπορεί να έχει μέχρι 2^{32} θέσεις μνήμης. Εμείς θα θεωρήσουμε ότι έχει 2^{32} θέσεις. Το μήκος κάθε θέσης είναι μία λέξη, δηλαδή 16 δυαδικά ψηφία. Το μέγεθος του μπλοκ της κύριας μνήμης είναι το ίδιο με το μέγεθος του πλαισίου της κρυφής μνήμης, επομένως η κύρια μνήμη έχει $2^{32}/2^2 = 2^{30}$ μπλοκ και στο ίδιο σύνολο της κρυφής μνήμης μπορούν να αποθηκευτούν $2^{30}/2^{12} = 2^{30-12} = 2^{18}$ μπλοκ της κύριας μνήμης.

β. $960F8_{(16)} = 1001\ 0110\ 0000\ 1111\ 1000_{(2)}$. Επειδή κάθε μπλοκ αποτελείται από 4 λέξεις, τα δύο λιγότερο σημαντικά δυαδικά ψηφία της διεύθυνσης δηλώνουν τη διεύθυνση της λέξης μέσα στο μπλοκ και μέσα στο πλαίσιο. Επειδή έχουμε 2^{12} σύνολα τα επόμενα 12 δυαδικά ψηφία δηλώνουν τη διεύθυνση του συνόλου. Τα υπόλοιπα 18 πιο σημαντικά δυαδικά ψηφία δηλώνουν την ετικέτα (βλέπε σχήμα 5.8.1).

000000000000100101	100000111110	00
ή		
00025	83E	0
ετικέτα	διεύθυνση συνόλου	διεύθυνση μέσα στο πλαίσιο
Διεύθυνση μπλοκ		διεύθυνση μέσα στο μπλοκ

Σχήμα 5.8.1

Άρα η λέξη που είναι αποθηκευμένη στη θέση μνήμης με διεύθυνση $960F8_{(16)}$ μπορεί να μεταφερθεί σε οποιοδήποτε πλαίσιο του συνόλου με διεύθυνση $83E$ της κρυφής μνήμης.

γ. Τα μπλοκ της κύριας μνήμης που μπορούν να αποθηκευτούν στο σύνολο της κρυφής μνήμης με διεύθυνση $060_{(16)}$ είναι αυτά που έχουν διεύθυνση:

$$R \ W \ Z \ Y \ X \ 0 \ 6 \ 0$$

όπου τα W, Z, Y και X παίρνουν τιμές από 0 έως και $F_{(16)}$, ενώ το R παίρνει τιμές από 0 έως και 3.

Άσκηση 5.9 (5.6 βιβλίου)

Θεωρήστε κρυφή μνήμη των 256 πλαισίων με 16 λέξεις ανά πλαίσιο και οργάνωση:

- i. Μονοσήμαντης απεικόνισης
- ii. Πλήρους συσχέτισης
- iii. 16-τρόπων συνόλου συσχέτισης.

α. Να δώσετε στο δεκαεξαδικό τις διευθύνσεις των πλαισίων της κρυφής μνήμης στα οποία μπορούν να αποθηκευτούν τα περιεχόμενα των θέσεων της κύριας μνήμης με διευθύνσεις $228A46, 219A45, 777F12$ σε κάθε μία από τις περιπτώσεις i. ii. iii.

β. Ποια από τα περιεχόμενα των ανωτέρω θέσεων μνήμης είναι δυνατόν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη σε κάθε μία από τις περιπτώσεις i. ii. iii και γιατί;

Απάντηση

α.i Οργάνωση μονοσήμαντης απεικόνισης .

Αφού κάθε πλαίσιο αποτελείται από $16=2^4$ λέξεις τα 4 λιγότερο σημαντικά δυαδικά ψηφία της διεύθυνσης δηλώνουν τη διεύθυνση της λέξης, μέσα στο μπλόκ καθώς και μέσα στο πλαίσιο. Επειδή η κρυφή μνήμη έχει $256=2^8$ πλαίσια και οργάνωση μονοσήμαντης απεικόνισης, τα επόμενα 8 λιγότερο σημαντικά δυαδικά ψηφία δίνουν την διεύθυνση του πλαισίου. Λαμβάνοντας υπόψη ότι κάθε ψηφίο στο δεκαεξαδικό είναι των τεσσάρων δυαδικών ψηφίων έχουμε:

3 ψηφία στο δεκαεξαδικό	2 ψηφία στο δεκαεξαδικό	1 ψηφίο στο δεκαεξαδικό
ετικέτα	διεύθυνση πλαισίου	διεύθυνση της λέξης μέσα στο πλαίσιο
διεύθυνση μπλοκ στην κύρια μνήμη		διεύθυνση της λέξης μέσα στο μπλοκ

Στον επόμενο πίνακα δίνονται στο δεκαεξαδικό οι διευθύνσεις των πλαισίων της κρυφής μνήμης στα οποία μπορούν να αποθηκευτούν τα περιεχόμενα των θέσεων της κύριας μνήμης με διευθύνσεις $228A46_{(16)}$, $219A45_{(16)}$ και $777F12_{(16)}$.

διεύθυνση θέσης μνήμης	διεύθυνση πλαισίου της κρυφής μνήμης
$228A46_{(16)}$	$A4_{(16)}$
$219A45_{(16)}$	$A4_{(16)}$
$777F12_{(16)}$	$F1_{(16)}$

α.ii Οργάνωση πλήρους συνόλου συσχέτισης.

Στην οργάνωση πλήρους συσχέτισης οποιοδήποτε μπλοκ της κύριας μνήμης μπορεί να μεταφερθεί σε οποιοδήποτε πλαίσιο της κρυφής μνήμης. Επομένως κάθε μια από τις διευθύνσεις μπορεί να αποθηκευτεί σε ένα πλαίσιο με διεύθυνση από 00 μέχρι και FF.

α.iii Οργάνωση 16-τρόπων συνόλου συσχέτισης.

Σ' αυτή την περίπτωση έχουμε 16 πλαίσια ανά σύνολο, άρα η κρυφή μνήμη θα έχει $256/16$ σύνολα = 16 σύνολα. Αφού κάθε πλαίσιο έχει $16=2^4$ λέξεις, τα 4 λιγότερο σημαντικά δυαδικά ψηφία της διεύθυνσης δηλώνουν και πάλι τη διεύθυνση της λέξης μέσα στο μπλόκ και μέσα στο πλαίσιο. Επομένως τα επόμενα $\log_2 16 = 4$ λιγότερο σημαντικά δυαδικά ψηφία θα δίνουν την διεύθυνση του συνόλου. Λαμβάνοντας υπόψη ότι κάθε ψηφίο στο δεκαεξαδικό είναι των τεσσάρων δυαδικών ψηφίων έχουμε:

4 ψηφία στο δεκαεξαδικό	1 ψηφίο στο δεκαεξαδικό	1 ψηφίο στο δεκαεξαδικό
ετικέτα	διεύθυνση συνόλου	διεύθυνση της λέξης μέσα στο πλαίσιο
διεύθυνση μπλοκ στην κύρια μνήμη		διεύθυνση της λέξης μέσα στο μπλοκ

Στον επόμενο πίνακα δίνονται στο δεκαεξαδικό οι διευθύνσεις των συνόλων της κρυφής μνήμης στα οποία μπορούν να αποθηκευτούν τα περιεχόμενα των θέσεων της κύριας μνήμης με διευθύνσεις $228A46_{(16)}$, $219A45_{(16)}$ και $777F12_{(16)}$.

Διεύθυνση θέσης μνήμης	Διεύθυνση συνόλου της κρυφής μνήμης
$228A46_{(16)}$	4
$219A45_{(16)}$	4
$777F12_{(16)}$	1

β.i Οργάνωση μονοσήμαντης απεικόνισης

Οι διευθύνσεις 228-A4-6 και 219-A4-5 ανήκουν σε διαφορετικά μπλοκ της κύριας μνήμης με διεύθυνση 228A4 και 219A4 αντίστοιχα, τα οποία όμως αποθηκεύονται στο ίδιο πλαίσιο με διεύθυνση A4. Επομένως τα περιεχόμενα των θέσεων της κύριας μνήμης με διευθύνσεις 228A46 και 219A45 δεν μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη.

Η διεύθυνση 777-F1-2 ανήκει στο μπλοκ με διεύθυνση 777F1 που αποθηκεύεται στο πλαίσιο F1 που είναι διαφορετικό από το πλαίσιο στο οποίο αποθηκεύονται τα μπλοκ 228A4 και 219A4, άρα τα περιεχόμενα της θέσης με διεύθυνση 777F12 μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη με τα περιεχόμενα των θέσεων με διευθύνσεις 228A46 ή 219A45.

ii Οργάνωση πλήρους συνόλου συσχέτισης

Σε κρυφή μνήμη πλήρους συνόλου συσχέτισης οποιοσδήποτε συνδυασμός μπλοκ της κύριας μνήμης που δεν ξεπερνά το πλήθος των πλαισίων της κρυφής μνήμης μπορεί να βρίσκεται ταυτόχρονα στην κρυφή μνήμη. Επειδή η κρυφή μνήμη έχει 256 πλαίσια, τα περιεχόμενα των θέσεων της κύριας μνήμης και των τριών διευθύνσεων που δόθηκαν, μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη.

β.iii Οργάνωση 16-τρόπων συνόλου συσχέτισης

Επειδή κάθε σύνολο έχει 16 πλαίσια, έως και 16 μπλοκ της κύριας μνήμης τα οποία αποθηκεύονται στο ίδιο σύνολο, μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη. Αφού εμείς έχουμε τρεις διευθύνσεις, συνεπάγεται ότι τα περιεχόμενα των θέσεων της κύριας μνήμης και των τριών διευθύνσεων που δόθηκαν, μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη, άσχετα αν αποθηκεύονται ή όχι στο ίδιο σύνολο.

Άσκηση 5.10 (5.7 βιβλίου)

Θεωρήστε τρεις κρυφές μνήμες των 512 πλαισίων με 8 λέξεις ανά πλαίσιο και οργάνωσεις αντίστοιχα:

- i. Μονοσήμαντης απεικόνισης
- ii. Πλήρους συσχέτισης
- iii. 8-τρόπων συνόλου συσχέτισης.

α. Να δώσετε στο οκταδικό τις διευθύνσεις των πλαισίων της κρυφής μνήμης στα οποία μπορούν να αποθηκευτούν τα περιεχόμενα των θέσεων της κύριας μνήμης με διευθύνσεις $117165_{(8)}$, $117167_{(8)}$, $445231_{(8)}$, $575232_{(8)}$, $675253_{(8)}$ και $677335_{(8)}$ σε κάθε μία από τις περιπτώσεις i. ii. iii.

β. Ποια από τα περιεχόμενα των ανωτέρω θέσεων μνήμης είναι δυνατόν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη σε κάθε μία από τις περιπτώσεις i. ii. iii και γιατί;

Απάντηση

α.i. Οργάνωση μονοσήμαντης απεικόνισης

Αφού κάθε πλαίσιο έχει $8=2^3$ λέξεις, τα 3 λιγότερο σημαντικά δυαδικά ψηφία της διεύθυνσης θα καθορίζουν τη θέση της λέξης μέσα στο μπλοκ της κύριας μνήμης και το πλαίσιο της κρυφής μνήμης.

Αφού η κρυφή μνήμη έχει $512=2^9$ πλαίσια, τα αμέσως επόμενα 9 λιγότερο σημαντικά δυαδικά ψηφία της διεύθυνσης θα καθορίζουν την διεύθυνση του πλαισίου στην κρυφή μνήμη. Λαμβάνοντας υπόψη ότι κάθε ψηφίο στο οκταδικό είναι των 3 δυαδικών ψηφίων έχουμε:

2 ψηφία στο οκταδικό	3 ψηφία στο οκταδικό	1 ψηφίο στο οκταδικό
Ετικέτα	Διεύθυνση πλαισίου	Διεύθυνση της λέξης μέσα στο πλαίσιο
Διεύθυνση μπλοκ στην κύρια μνήμη		Διεύθυνση της λέξης μέσα στο μπλοκ

Στον επόμενο πίνακα δίνονται στο οκταδικό οι διευθύνσεις των πλαισίων της κρυφής μνήμης στα οποία μπορούν να αποθηκευτούν τα περιεχόμενα των θέσεων της κύριας μνήμης με διευθύνσεις $117165_{(8)}$, $117167_{(8)}$, $445231_{(8)}$, $575232_{(8)}$, $675253_{(8)}$ και $677335_{(8)}$.

Διεύθυνση θέσης μνήμης	Διεύθυνση πλαισίου της κρυφής μνήμης
$117165_{(8)}$	$716_{(8)}$
$117167_{(8)}$	$716_{(8)}$
$445231_{(8)}$	$523_{(8)}$
$575232_{(8)}$	$523_{(8)}$
$675253_{(8)}$	$525_{(8)}$
$677335_{(8)}$	$733_{(8)}$

α.ii. Οργάνωση πλήρους συνόλου συσχέτισης.

Στην οργάνωση πλήρους συνόλου συσχέτισης οποιοδήποτε μπλοκ της κύριας μνήμης μπορεί να μεταφερθεί σε οποιοδήποτε πλαίσιο της κρυφής μνήμης. Οι διευθύνσεις $117165_{(8)}$ και $117167_{(8)}$ ανήκουν στο ίδιο μπλοκ, επομένως μπορούν να μεταφερθούν σε οποιοδήποτε πλαίσιο, αλλά και οι δύο πάντα στο ίδιο πλαίσιο, της κρυφής μνήμης. Κάθε μία από τις υπόλοιπες διευθύνσεις μπορεί να μεταφερθεί σε οποιοδήποτε πλαίσιο της κρυφής μνήμης.

α.iii. Οργάνωση 8-τρόπων συνόλου συσχέτισης.

Σ' αυτή την περίπτωση έχουμε 8 πλαίσια ανά σύνολο, επομένως έχουμε $512/8$ σύνολα = 64 σύνολα. Αφού κάθε πλαίσιο έχει $8=2^3$ λέξεις, τα 3 λιγότερο σημαντικά δυαδικά ψηφία της διεύθυνσης θα καθορίζουν και πάλι τη θέση της λέξης μέσα στο μπλοκ της κύριας μνήμης και το πλαίσιο της κρυφής μνήμης.

Αφού η κρυφή μνήμη έχει $64=2^6$ σύνολα, τα αμέσως επόμενα 6 λιγότερο σημαντικά δυαδικά ψηφία της διεύθυνσης θα καθορίζουν την διεύθυνση του συνόλου στην κρυφή μνήμη. Λαμβάνοντας υπόψη ότι κάθε ψηφίο στο οκταδικό είναι των 3 δυαδικών ψηφίων έχουμε:

3 ψηφία στο οκταδικό	2 ψηφία στο οκταδικό	1 ψηφίο στο οκταδικό
ετικέτα	διεύθυνση συνόλου	διεύθυνση της λέξης μέσα στο πλαίσιο
διεύθυνση μπλοκ στην κύρια μνήμη		διεύθυνση της λέξης μέσα στο μπλοκ

Στον επόμενο πίνακα δίνονται στο οκταδικό οι διευθύνσεις των συνόλων της κρυφής μνήμης στα οποία μπορούν να αποθηκευτούν τα περιεχόμενα των θέσεων της κύριας μνήμης με διευθύνσεις $117165_{(8)}$, $117167_{(8)}$, $445231_{(8)}$, $575232_{(8)}$, $675253_{(8)}$ και $677335_{(8)}$.

Διεύθυνση μνήμης	Διεύθυνση συνόλου της κρυφής μνήμης
$117165_{(8)}$	$16_{(8)}$
$117167_{(8)}$	$16_{(8)}$
$445231_{(8)}$	$23_{(8)}$
$575232_{(8)}$	$23_{(8)}$
$675253_{(8)}$	$25_{(8)}$
$677335_{(8)}$	$33_{(8)}$

β.i. Οργάνωση μονοσήμαντης απεικόνισης

Οι διευθύνσεις $11-716-5_{(8)}$ και $11-716-7_{(8)}$ ανήκουν στο ίδιο μπλοκ (έχουν την ίδια ετικέτα) επομένως όταν βρίσκονται τα περιεχόμενα της μίας διεύθυνσης στην κρυφή μνήμη θα βρίσκονται και της άλλης.

Οι διευθύνσεις $44-523-1_{(8)}$ και $57-523-2_{(8)}$ ανήκουν σε διαφορετικά μπλοκ της κύριας μνήμης που αντιστοιχούν στο ίδιο πλαίσιο της κρυφής μνήμης με διεύθυνση 523, επομένως τα περιεχόμενά τους δεν μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη.

Οι διευθύνσεις 67-525-3₍₈₎ και 67-733-5₍₈₎ μεταξύ τους και σε σχέση με τις προηγούμενες διευθύνσεις ανήκουν σε διαφορετικά μπλοκ της κύριας μνήμης που αντιστοιχούν σε διαφορετικά πλαίσια της κρυφής μνήμης επομένως τα περιεχόμενά τους μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη.

Άρα στην κρυφή μνήμη μπορούν να βρίσκονται ταυτόχρονα τα περιεχόμενα των θέσεων της κύριας μνήμης με διευθύνσεις 117165₍₈₎, 117167₍₈₎, 445231₍₈₎, 675253₍₈₎ και 677335₍₈₎ ή 117165₍₈₎, 117167₍₈₎, 575232₍₈₎, 675253₍₈₎ και 677335₍₈₎.

β.ii. Οργάνωση πλήρους συνόλου συσχέτισης

Στην οργάνωση πλήρους συνόλου συσχέτισης οποιοσδήποτε συνδυασμός μπλοκ μικρότερος του πλήθους των πλαισίων μπορεί να βρίσκεται ταυτόχρονα στην κρυφή μνήμη. Αφού στην περίπτωση μας έχουμε μόνο 6 διευθύνσεις κύριας μνήμης και 512 πλαίσια, τα περιεχόμενα των διευθύνσεων αυτών μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη.

β.iii. Οργάνωση 8-τρόπων συνόλου συσχέτισης

Αφού έχουμε μόνο 6 διευθύνσεις κύριας μνήμης και 8 πλαίσια ανά σύνολο, τα περιεχόμενα των διευθύνσεων αυτών θα μπορούσαν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη ακόμη και αν όλα αντιστοιχούσαν στο ίδιο σύνολο (πράγμα βέβαια που δεν συμβαίνει).

Άσκηση 5.11 (5.8 βιβλίου)

Θεωρήστε ότι η ΚΜΕ ενός υπολογιστή, που παράγει διευθύνσεις των 7 δυαδικών ψηφίων, παράγει την επόμενη ακολουθία διευθύνσεων: 33, 17, 30, 47, 50, 35, 6, 26, 50, 42, 58, 50, 13, 22, 15, 4, 0, 70. Για κάθε μία από τις επόμενες περιπτώσεις να γράψετε δίπλα σε κάθε διεύθυνση εάν έχουμε επιτυχία ή αποτυχία και στη συνέχεια να δώσετε το τελικό περιεχόμενο της κρυφής μνήμης. Στο ξεκίνημα η κρυφή μνήμη δεν έχει έγκυρα περιεχόμενα.

α. Η κρυφή μνήμη έχει οργάνωση μονοσήμαντης απεικόνισης με χωρητικότητα 32 λέξεων και 4 λέξεις ανά πλαίσιο.

β. Η κρυφή μνήμη έχει οργάνωση 2-τρόπων συνόλου συσχέτισης, χωρητικότητα 32 λέξεων και 2 λέξεις ανά πλαίσιο. Υποθέστε ότι χρησιμοποιείται η στρατηγική απελευθέρωσης πλαισίων της κρυφής μνήμης “FIFO”.

γ. Η κρυφή μνήμη έχει οργάνωση πλήρους συνόλου συσχέτισης, χωρητικότητα 32 λέξεων και 4 λέξεις ανά πλαίσιο. Υποθέστε ότι χρησιμοποιείται η στρατηγική απελευθέρωσης πλαισίων της κρυφής μνήμης “LRU”.

Απάντηση

α. Η κρυφή μνήμη έχει οργάνωση μονοσήμαντης απεικόνισης με χωρητικότητα 32 λέξεων και 4 λέξεις ανά πλαίσιο, άρα έχει $32/4 = 8$ πλαίσια των τεσσάρων λέξεων. Επομένως τα 2 λιγότερο σημαντικά δυαδικά ψηφία χρησιμοποιούνται για τον καθορισμό της θέσης της λέξης μέσα στο πλαίσιο, τα επόμενα 3 δυαδικά ψηφία χρησιμοποιούνται για τον καθορισμό του πλαισίου και τα υπόλοιπα δύο δυαδικά ψηφία αποτελούν την ετικέτα. Στον Πίνακα 5.11.1 περιγράφουμε τι θα συμβεί για κάθε διεύθυνση που παράγει η ΚΜΕ.

Πίνακας 5.11.1
Οργάνωση μονοσήμαντης απεικόνισης

διεύθυνση μνήμης (δεκαδικό)	διεύθυνση σε δυαδική μορφή (ετικέτα – πλαίσιο – λέξη)	επιτυχία ή αποτυχία (επ ή απ)	μεταφορά δεδομένων στην κρυφή μνήμη
33	01-000-01	απ	ΜΠ(32, 33, 34, 35) => Π0*
17	00-100-01	απ	ΜΠ(16, 17, 18, 19) => Π4
30	00-111-10	απ	ΜΠ(28, 29, 30, 31) => Π7
47	01-011-11	απ	ΜΠ(44, 45, 46, 47) => Π3
50	01-100-10	απ	ΜΠ(48, 49, 50, 51) => Π4
35	01-000-11	επ	
6	00-001-10	απ	ΜΠ(4, 5, 6, 7) => Π1
26	00-110-10	απ	ΜΠ(24, 25, 26, 27) => Π6
50	01-100-10	επ	
42	01-010-10	απ	ΜΠ(40, 41, 42, 43) => Π2
58	01-110-10	απ	ΜΠ(56, 57, 58, 59) => Π6
50	01-100-10	επ	
13	00-011-01	απ	ΜΠ(12, 13, 14, 15) => Π3

* Ο συμβολισμός ΜΠ(X,...,Y) => ΠZ δηλώνει ότι το περιεχόμενο του μπλοκ με διευθύνσεις της κυρίας μνήμης από X έως και Y αποθηκεύεται στο πλαίσιο Z.

22	00-101-10	απ	ΜΠ(20, 21, 22, 23) => Π5
15	00-011-11	επ	
4	00-001-00	επ	
0	00-000-00	απ	ΜΠ(0, 1, 2, 3) => Π0
70	10-001-10	απ	ΜΠ(68, 69, 70, 71) => Π1

Μετά από τη συγκεκριμένη ακολουθία διευθύνσεων, τα περιεχόμενα της κρυφής μνήμης δίνονται στον Πίνακα 5.11.2.

Πίνακας 5.11.2

Περιεχόμενα της κρυφής μνήμης με οργάνωση μονοσήμαντης απεικόνισης

Πλαίσιο 0	ΜΠ(0, 1, 2, 3)
Πλαίσιο 1	ΜΠ(68, 69, 70, 71)
Πλαίσιο 2	ΜΠ(40, 41, 42, 43)
Πλαίσιο 3	ΜΠ(12, 13, 14, 15)
Πλαίσιο 4	ΜΠ(48, 49, 50, 51)
Πλαίσιο 5	ΜΠ(20, 21, 22, 23)
Πλαίσιο 6	ΜΠ(56, 57, 58, 59)
Πλαίσιο 7	ΜΠ(28, 29, 30, 31)

β. Η κρυφή μνήμη έχει οργάνωση 2-τρόπων συνόλου συσχέτισης, χωρητικότητα 32 λέξεων και δύο λέξεις ανά πλαίσιο. Άρα κάθε σύνολο έχει δύο πλαίσια των δύο λέξεων, συνεπώς η κρυφή μνήμη περιλαμβάνει οκτώ σύνολα. Επομένως το 1 λιγότερο σημαντικό δυαδικό ψηφίο καθορίζει τη διεύθυνση της λέξης μέσα στο πλαίσιο, τα επόμενα 3 δυαδικά ψηφία καθορίζουν τη διεύθυνση του συνόλου και τα υπόλοιπα 3 πιο σημαντικά δυαδικά ψηφία αποτελούν την ετικέτα. Στον Πίνακα 5.11.3 περιγράφουμε τι θα συμβεί για κάθε διεύθυνση που παράγει η ΚΜΕ.

Πίνακας 5.11.3
Οργάνωση 2-τρόπων συνόλου συσχέτισης

διεύθυνση μνήμης (δεκαδικό)	διεύθυνση (ετικέτα – σύνολο - λέξη)	επιτυχία ή αποτυχία (επ ή απ)	μεταφορά δεδομένων στην κρυφή μνήμη	
			πλαίσιο 0	πλαίσιο 1
33	010-000-1	απ	ΜΠ(32, 33) => Σ0*	
17	001-000-1	απ		ΜΠ(16, 17) => Σ 0
30	001-111-0	απ	ΜΠ(30, 31) => Σ7	
47	010-111-1	απ		ΜΠ(46, 47) => Σ7
50	011-001-0	απ	ΜΠ(50, 51) => Σ1	
35	010-001-1	απ		ΜΠ(34, 35) => Σ1
6	000-011-0	απ	ΜΠ(6, 7) => Σ3	
26	001-101-0	απ	ΜΠ(26, 27) => Σ5	
50	011-001-0	επ		
42	010-101-0	απ		ΜΠ(42, 43) => Σ5
58	011-101-0	απ	ΜΠ(58, 59) => Σ5 (FIFO)	
50	011-001-0	επ		
13	000-110-1	απ	ΜΠ(12, 13) => Σ6	
22	001-011-0	απ		ΜΠ(22, 23) => Σ3
15	000-111-1	απ	ΜΠ(14, 15) => Σ7 (FIFO)	
4	000-010-0	απ	ΜΠ(4, 5) => Σ2	
0	000-000-0	απ	ΜΠ(0, 1) => Σ0 (FIFO)	
70	100-011-0	απ	ΜΠ(70, 71) => Σ3 (FIFO)	

* Ο συμβολισμός ΜΠ(X, ... ,Y) => ΣZ δηλώνει ότι το περιεχόμενο του μπλοκ με διευθύνσεις της κυρίας μνήμης από X έως και Y αποθηκεύεται σε ένα πλαίσιο του συνόλου Z.

Μετά από τη συγκεκριμένη ακολουθία διευθύνσεων, τα περιεχόμενα της κρυφής μνήμης δίνονται στον Πίνακα 5.11.4.

Πίνακας 5.11.4

Περιεχόμενα της κρυφής μνήμης με οργάνωση 2-τρόπων συνόλου συσχέτισης

	Πλαίσιο 0	Πλαίσιο 1
Σύνολο 0	ΜΠ(0, 1)	ΜΠ(16, 17)
Σύνολο 1	ΜΠ(50, 51)	ΜΠ(34, 35)
Σύνολο 2	ΜΠ(4, 5)	-
Σύνολο 3	ΜΠ(70, 71)	ΜΠ(22, 23)
Σύνολο 4	-	-
Σύνολο 5	ΜΠ(58, 59)	ΜΠ(42, 43)
Σύνολο 6	ΜΠ(12, 13)	-
Σύνολο 7	ΜΠ(14, 15)	ΜΠ(46, 47)

Η παύλα σημαίνει ότι το συγκεκριμένο πλαίσιο δεν έχει έγκυρα περιεχόμενα.

γ. Η κρυφή μνήμη έχει οργάνωση πλήρους συσχέτισης, χωρητικότητα 32 λέξεων και τέσσερις λέξεις ανά πλαίσιο. Άρα η κρυφή μνήμη έχει οκτώ πλαίσια των τεσσάρων λέξεων. Από τα επτά ψηφία της διεύθυνσης τα δύο λιγότερο σημαντικά δυαδικά ψηφία χρησιμοποιούνται για να καθορίσουν τη θέση της λέξης μέσα σε ένα πλαίσιο και τα υπόλοιπα πέντε δυαδικά ψηφία αποτελούν την ετικέτα, η οποία δηλώνει τη διεύθυνση του μπλοκ που έχει αποθηκευτεί σε συγκεκριμένο πλαίσιο. Στον Πίνακα 5.11.5 περιγράφουμε τι θα συμβεί για κάθε διεύθυνση που παράγει η ΚΜΕ.

Πίνακας 5.11.5
Οργάνωση πλήρους συσχέτισης

διεύθυνση μνήμης (δεκαδικό)	Διεύθυνση (ετικέτα –λέξη)	επιτυχία ή αποτυχία (επ ή απ)	μεταφορά δεδομένων στην κρυφή μνήμη
33	01000-01	απ	ΜΠ(32, 33, 34, 35) => Π0*
17	00100-01	απ	ΜΠ(16, 17, 18, 19) => Π1
30	00111-10	απ	ΜΠ(28, 29, 30, 31) => Π2
47	01011-11	απ	ΜΠ(44, 45, 46, 47) => Π3
50	01100-10	απ	ΜΠ(48, 49, 50, 51) => Π4
35	01000-11	επ	
6	00001-10	απ	ΜΠ(4, 5, 6, 7) => Π5
26	00110-10	απ	ΜΠ(24, 25, 26, 27) => Π6
50	01100-10	επ	
42	01010-10	απ	ΜΠ(40, 41, 42, 43) => Π7
58	01110-10	απ	ΜΠ(56, 57, 58, 59) => Π1 (LRU)
50	01100-10	επ	
13	00011-01	απ	ΜΠ(12, 13, 14, 15) => Π2 (LRU)
22	00101-10	απ	ΜΠ(20, 21, 22, 23) => Π3 (LRU)
15	00011-11	επ	
4	00001-00	επ	
0	00000-00	απ	ΜΠ(0, 1, 2, 3) => Π0 (LRU)
70	10001-10	απ	ΜΠ(68, 69, 70, 71) => Π6 (LRU)

* Ο συμβολισμός ΜΠ(X, ... ,Y) => ΠΖ δηλώνει ότι το περιεχόμενο του μπλοκ με διευθύνσεις της κυρίας μνήμης από X έως και Y αποθηκεύεται στο πλαίσιο Ζ.

Μετά από τη συγκεκριμένη ακολουθία διευθύνσεων, τα περιεχόμενα της κρυφής μνήμης δίνονται στον Πίνακα 5.11.6.

Πίνακας 5.11.6

Περιεχόμενα της κρυφής μνήμης με οργάνωση πλήρους συσχέτισης

Διεύθυνση πλαισίου	Διευθύνσεις θέσεων της κύριας μνήμης τα περιεχόμενα των οποίων βρίσκονται στο συγκεκριμένο πλαίσιο
Πλαίσιο 0	ΜΠ(0, 1, 2, 3)
Πλαίσιο 1	ΜΠ(56, 57, 58, 59)
Πλαίσιο 2	ΜΠ(12, 13, 14, 15)
Πλαίσιο 3	ΜΠ(20, 21, 22, 23)
Πλαίσιο 4	ΜΠ(48, 49, 50, 51)
Πλαίσιο 5	ΜΠ(4, 5, 6, 7)
Πλαίσιο 6	ΜΠ(68, 69, 70, 71)
Πλαίσιο 7	ΜΠ(40, 41, 42, 43)

Άσκηση 5.12 (5.9 βιβλίου)

Θεωρήστε ότι σε ένα υπολογιστικό σύστημα, το οποίο δεν έχει ιδεατή μνήμη, υπάρχει μία κρυφή μνήμη δεδομένων με οργάνωση τ-τρόπων συνόλου συσχέτισης. Η κρυφή μνήμη αποτελείται από 64 πλαίσια και κάθε πλαίσιο της κρυφής μνήμης είναι των τεσσάρων λέξεων (στην περίπτωση μας μία λέξη = 16 δυαδικά ψηφία).

α. Προσδιορίστε τη μικρότερη τιμή του τ για την οποία έχουμε το μικρότερο αριθμό αποτυχιών (misses) στην κρυφή μνήμη κατά την εκτέλεση του προγράμματος που ακολουθεί. Σημειώστε ότι για $\tau=1$ έχουμε κρυφή μνήμη με οργάνωση μονοσήμαντης απεικόνισης.

Πρόγραμμα.

LOAD D, # 05FF	/εντολή 1
LOAD H, # 02FF	/εντολή 2
LOOP INCR D	/εντολή 3
INCR H	/εντολή 4
LOAD A, [D]	/εντολή 5
ADD A, [H]	/εντολή 6
STORE A, [H]	/εντολή 7
MOVE A, H	/εντολή 8
COMP A, #03FF	/εντολή 9
JNZ LOOP	/εντολή 10
END	/εντολή 11

Σημείωση:

LOAD register, # number: αποθήκευσε στον καταχωρητή τον αριθμό

INCR register: αύξησε το περιεχόμενο του καταχωρητή κατά ένα

LOAD register1, [register2]: αποθήκευσε στον καταχωρητή 1 το περιεχόμενο της θέσης μνήμης που η διεύθυνσή της δίνεται από το περιεχόμενο του καταχωρητή 2

STORE register1, [register2]: αποθήκευσε το περιεχόμενο του καταχωρητή 1 στη θέση μνήμης που η διεύθυνσή της δίνεται από το περιεχόμενο του καταχωρητή 2

ADD register1, [register2]: πρόσθεσε το περιεχόμενο του καταχωρητή 1 με το περιεχόμενο της θέσης μνήμης που η διεύθυνσή της δίνεται από το περιεχόμενο του καταχωρητή 2 και βάλε το αποτέλεσμα στον καταχωρητή 1.

MOVE register1, register2: αποθήκευσε στον καταχωρητή 1 το περιεχόμενο του καταχωρητή 2.

COMP register, # number: συνέκρινε το περιεχόμενο του καταχωρητή με τον αριθμό κάνοντας την αφαίρεση περιεχόμενο καταχωρητή μείον αριθμός

JNZ LOOP: εάν το αποτέλεσμα της προηγούμενης αριθμητικής πράξης δεν είναι μηδενικό μεταφέρεται ο έλεγχος στην εντολή με ετικέτα το loop.

β. Στην κρυφή μνήμη που επιλέξατε ποιος είναι ο αριθμός των αποτυχιών (misses) κατά την εκτέλεση του προηγούμενου προγράμματος;

Απάντηση

α. Το πρόγραμμα της άσκησης προσθέτει κάθε ένα από τους αριθμούς των 16 δυαδικών ψηφίων που είναι αποθηκευμένοι στις θέσεις μνήμης από $300_{(16)}$ έως και $3FF_{(16)}$ με τους αντίστοιχους αριθμούς που είναι αποθηκευμένοι στις θέσεις μνήμης από $600_{(16)}$ έως και $6FF_{(16)}$ και αποθηκεύει τα αθροίσματα στις θέσεις από $300_{(16)}$ έως και $3FF_{(16)}$. Με άλλα λόγια από το πρόγραμμα μπορούμε να παρατηρήσουμε εύκολα ότι η εντολή 6 προσθέτει τα περιεχόμενα των θέσεων μνήμης $00000011\theta\eta\zeta\epsilon\delta\gamma\beta_{(2)}$ και $00000110\theta\eta\zeta\epsilon\delta\gamma\beta_{(2)}$ και η εντολή 7 αποθηκεύει το αποτέλεσμα στη θέση $00000011\theta\eta\zeta\epsilon\delta\gamma\beta_{(2)}$ για $\theta\eta\zeta\epsilon\delta\gamma\beta$ από $00000000_{(2)}$ έως και $11111111_{(2)}$.

Ας εξετάσουμε αρχικά τι θα γίνει αν $\tau=1$, δηλαδή η κρυφή μνήμη έχει οργάνωση μονοσήμαντης απεικόνισης. Αφού κάθε πλαίσιο είναι των 4 λέξεων τα δύο λιγότερο σημαντικά ψηφία της διεύθυνσης θα επιλέγουν τη λέξη μέσα σε ένα πλαίσιο. Επειδή η κρυφή μνήμη έχει 64 πλαίσια τα επόμενα 6 δυαδικά ψηφία της διεύθυνσης χρησιμο-

ποιούνται για τη διευθυνσιοδότηση του πλαισίου και τα υπόλοιπα θα αποτελούν την ετικέτα. Τότε οι διευθύνσεις 00000011-θηζεδγ-βα₍₂₎ και 00000110-θηζεδγ-βα₍₂₎, τα περιεχόμενα των οποίων πρέπει να προστεθούν, αντιστοιχούν σε διαφορετικά μπλοκ της κύριας μνήμης αλλά στο ίδιο πλαίσιο της κρυφής μνήμης και επομένως δεν μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη.

Κάθε φορά που θα εκτελείται η εντολή 5 και η εντολή 6 του προγράμματος θα έχουμε αποτυχία. Λαμβάνοντας υπόψη ότι κάθε μία από τις εντολές αυτές εκτελείται 100₍₁₆₎ φορές, συμπεραίνουμε ότι θα έχουμε 200₍₁₆₎ αποτυχίες (misses).

Από την ανωτέρω ανάλυση προκύπτει ότι πρέπει να έχουμε τη δυνατότητα να κρατάμε ταυτόχρονα στην κρυφή μνήμη δύο μπλοκ της κύριας μνήμης, το μπλοκ με διεύθυνση 00000011-θηζεδγ₍₂₎ και το μπλοκ με διεύθυνση 00000110-θηζεδγ₍₂₎. Ας εξετάσουμε λοιπόν τι θα γίνει αν $\tau=2$, δηλαδή η κρυφή μνήμη είναι 2-τρόπων συνόλου συσχέτισης. Τότε η κρυφή μνήμη έχει $64/2=32$ σύνολα επομένως τα 5 δυαδικά ψηφία της διεύθυνσης, μετά τα δύο λιγότερο σημαντικά δυαδικά ψηφία τα οποία και πάλι επιλέγουν τη λέξη μέσα σε ένα πλαίσιο, χρησιμοποιούνται για την διευθυνσιοδότηση του συνόλου και τα υπόλοιπα αποτελούν την ετικέτα. Αφού κάθε σύνολο έχει δύο πλαίσια τα δύο μπλοκ, το μπλοκ με διεύθυνση 00000011θ-ηζεδγ₍₂₎ και το μπλοκ με διεύθυνση 00000110θ-ηζεδγ₍₂₎, της κύριας μνήμης που χρειάζονται ταυτόχρονα στην κρυφή μνήμη μπορούν να κρατηθούν στα δύο πλαίσια του συνόλου με διεύθυνση ηζεδγ₍₂₎. Τώρα κάθε μπλοκ της κύριας μνήμης, το οποίο χρειάζεται, προσκομίζεται μόνο μία φορά στην κρυφή μνήμη που είναι και ότι καλύτερο μπορεί να επιτευχθεί. Επομένως η 2-τρόπων συνόλου συσχέτισης κρυφή μνήμη εξασφαλίζει τον ελάχιστο αριθμό αποτυχιών στην εκτέλεση του δοθέντος προγράμματος.

β. Αφού κάθε ένα από τα μπλοκ της κύριας μνήμης που απαιτούνται προσπελάσσεται μόνο μία φορά, αρκεί να υπολογίσουμε το πλήθος των μπλοκ. Όπως έχουμε ήδη αναφέρει το πρόγραμμα της άσκησης προσθέτει κάθε ένα από τους αριθμούς που είναι αποθηκευμένοι στις θέσεις μνήμης από 300₍₁₆₎ έως και 3FF₍₁₆₎, 100₍₁₆₎ αριθμοί, με τους αντίστοιχους αριθμούς που είναι αποθηκευμένοι στις θέσεις μνήμης από 600₍₁₆₎ έως και 6FF₍₁₆₎, άλλοι 100₍₁₆₎ αριθμοί. Συνολικά έχουμε $200_{(16)} = 512_{(10)}$ διευθύνσεις οπότε λαμβάνοντας υπόψη ότι κάθε μπλοκ είναι των 4 λέξεων έχουμε $512/4=128$ αποτυχίες (misses).

Άσκηση 5.13 (5.10 βιβλίου)

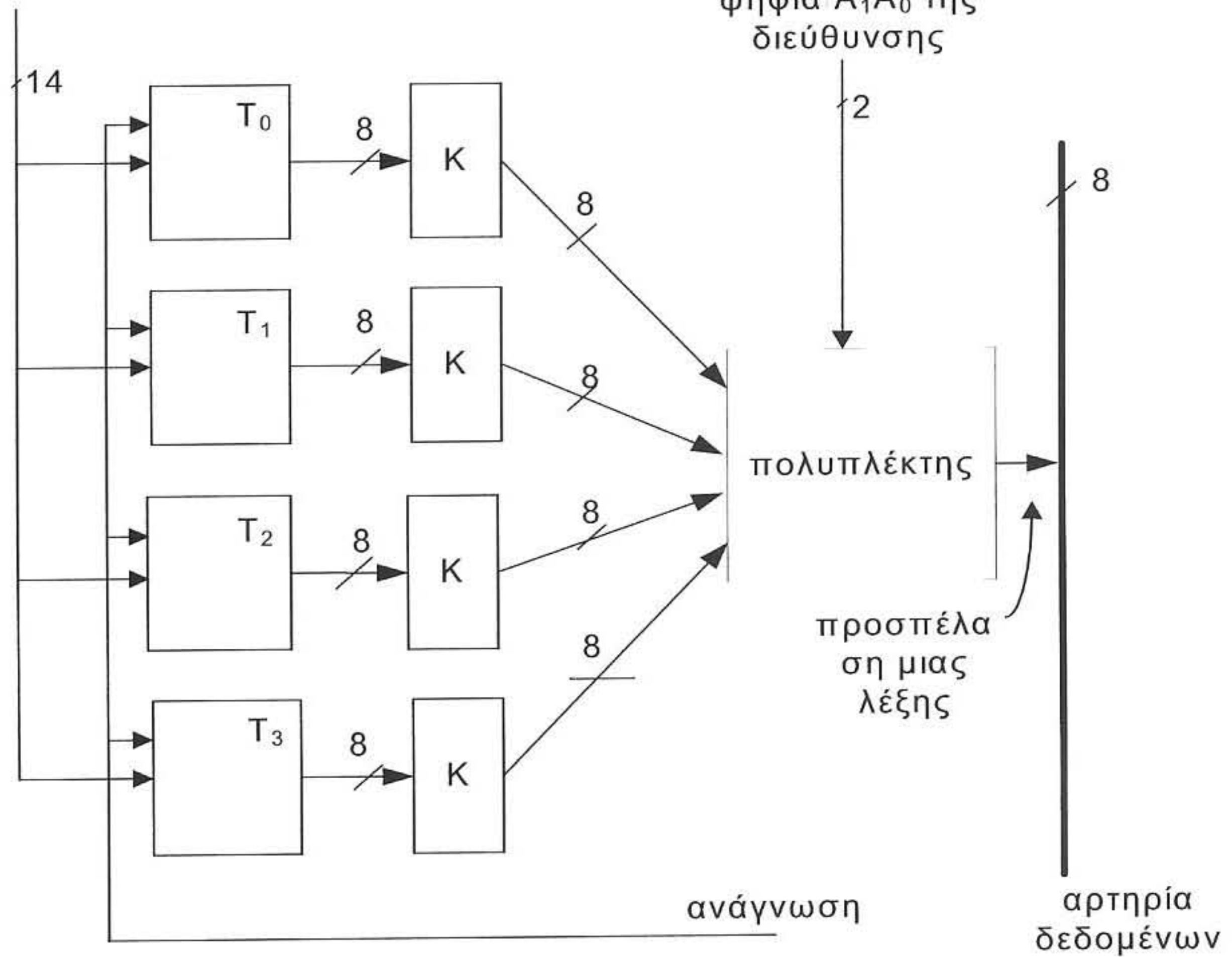
Θεωρήστε σύστημα μνήμης αποτελούμενο από κρυφή μνήμη δεδομένων και ξεχωριστή κρυφή μνήμη εντολών με 4 λέξεις της μίας ψηφιολέξης (byte) ανά πλαίσιο, κύρια μνήμη και βοηθητική μνήμη (το σύστημα δεν υποστηρίζει ιδεατή μνήμη). Κάθε θέση της κύριας μνήμης είναι των 8 δυαδικών ψηφίων. Η αρτηρία για την μεταφορά εντολών μεταξύ κρυφής μνήμης και κύριας μνήμης είναι των 8 δυαδικών ψηφίων. Η εγγραφή μίας ψηφιολέξης στην κρυφή μνήμη απαιτεί 1 κύκλο ρολογιού. Να σχεδιάσετε κύρια μνήμη με χωρητικότητα 64 Kψηφιολέξεις (Kbytes) έτσι ώστε η μεταφορά ενός μπλοκ της κύριας μνήμης σε ένα πλαίσιο της κρυφής μνήμης εντολών να ολοκληρώνεται σε 16 κύκλους ρολογιού. Έχετε στη διάθεσή σας ολοκληρωμένα κυκλώματα μνήμης DRAM με χωρητικότητα 8 Kψηφιολέξεις (Kbytes) με μία ψηφιολέξη ανά θέση μνήμης και χρόνο προσπέλασης ίσο με 12 κύκλους ρολογιού. Θεωρήστε ότι τα ολοκληρωμένα κυκλώματα μνήμης DRAM, τα οποία έχετε στη διάθεσή σας, δεν έχουν τη δυνατότητα γρήγορης προσπέλασης ομάδας θέσεων οι οποίες έχουν την ίδια διεύθυνση γραμμής. Θεωρήστε ότι έχετε στη διάθεσή σας οποιαδήποτε άλλα κυκλώματα χρειάζεστε και ότι η αρτηρία διευθύνσεων είναι των 16 δυαδικών ψηφίων.

Απάντηση

Τα δεδομένα ενός μπλοκ της κύριας μνήμης που πρέπει να μεταφερθούν σε ένα πλαίσιο της κρυφής μνήμης είναι αποθηκευμένα σε διαδοχικές διευθύνσεις. Αν στην οργάνωση της κύριας μνήμης αντιστοιχίσουμε τις διαδοχικές διευθύνσεις στο ίδιο ΟΚΜ, τότε αφού ο χρόνος προσπέλασης ενός ΟΚΜ είναι 12 κύκλοι ρολογιού, για να προσπελάσουμε 4 διαδοχικές διευθύνσεις απαιτούνται 4×12 κύκλοι ρολογιού που παραβιάζουν κατά πολύ τις απαιτήσεις μας. Αφού η αρτηρία εντολών είναι των 8 γραμμών και η εγγραφή μιας ψηφιολέξης στην κύρια μνήμη απαιτεί ένα κύκλο ρολογιού, συνεπάγεται ότι για την μεταφορά και εγγραφή των 4 λέξεων του πλαισίου στην κρυφή μνήμη απαιτούνται 4 κύκλοι ρολογιού. Επομένως, για το διάβασμα της κύριας μνήμης έχουμε στη διάθεσή μας $16 - 4 = 12$ κύκλους ρολογιού. Αφού η προσπέλαση της κύριας μνήμης απαιτεί 12 κύκλους ρολογιού, πρέπει να αντιστοιχίσουμε διαδοχικές διευθύνσεις σε διαφορετικά ΟΚΜ. Συγκεκριμένα αφού θέλουμε ανά 4 οι διαδοχικές διευθύνσεις να αντιστοιχούν σε διαφορετικά ΟΚΜ, θα χρησιμοποιήσουμε οργάνωση κύριας μνήμης 4-δρόμων χαμηλής τάξης διαφύλλωσης όπως φαίνεται στο σχήμα 5.13.1.

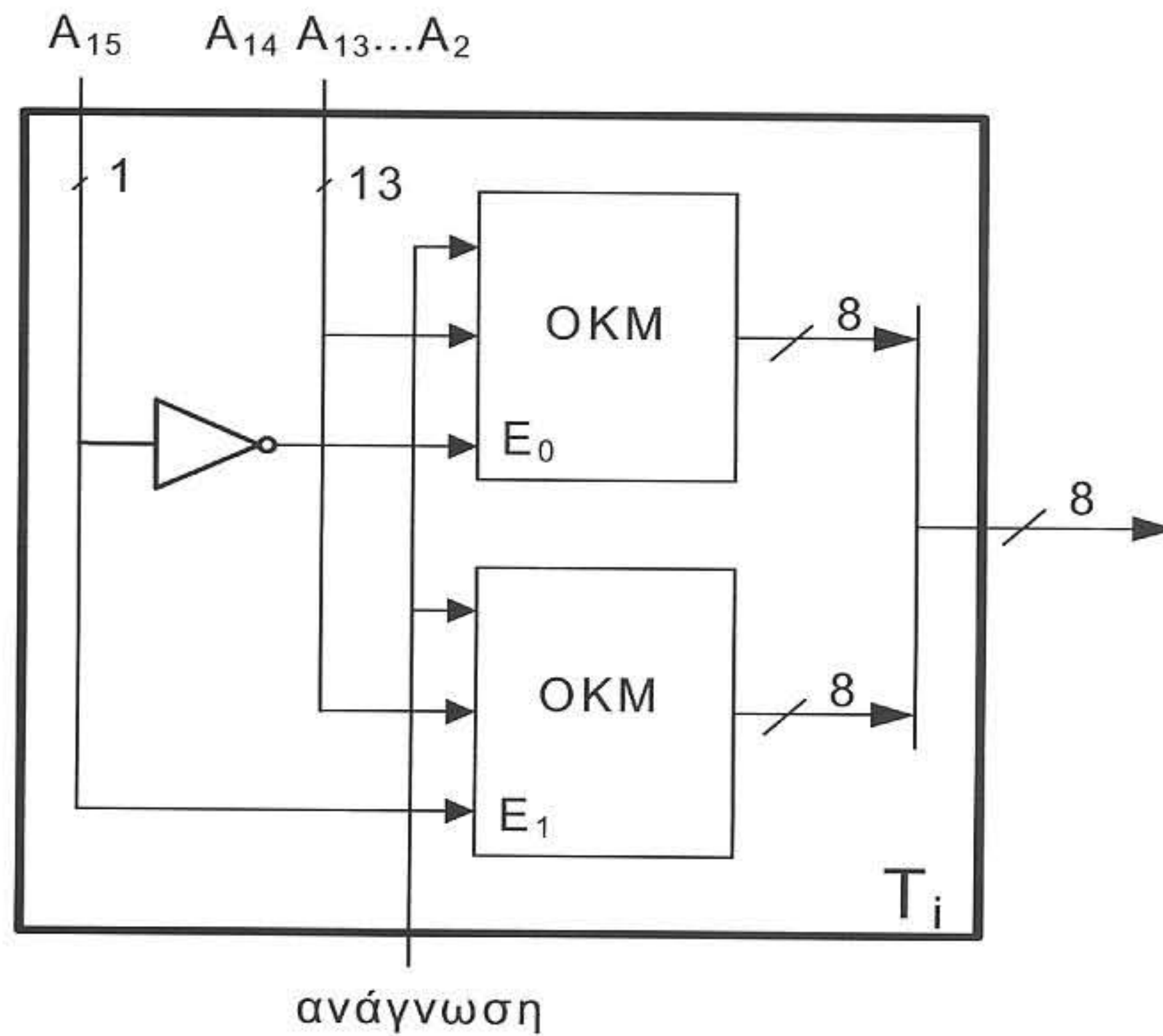
τα 14 αμέσως περισσότερα
σημαντικά δυαδικά ψηφία
 $A_{15} A_{14} \dots A_2$ της διεύθυνσης

τα 2 λιγότερο
σημαντικά δυαδικά
ψηφία $A_1 A_0$ της
διεύθυνσης



Σχήμα 5.13.1 Σύστημα μνήμης με οργάνωση 4- δρόμων χαμηλής τάξης διαφύλλωσης.

Τώρα τα 4 τμήματα προσπελούνται παράλληλα, άρα ο χρόνος προσπέλασης είναι 12 κύκλοι ρολογιού και το περιεχόμενο κάθε θέσης αποθηκεύεται στον αντίστοιχο καταχωρητή K. Σε κάθε ένα από τους επόμενους 4 κύκλους ρολογιού το περιεχόμενο ενός καταχωρητή αποθηκεύεται στην κρυφή μνήμη. Αφού θέλουμε 64 Kbytes κάθε τμήμα θα είναι των 16 Kbytes, οπότε αφού διαθέτουμε ΟΚΜ με χωρητικότητα 8 Kbytes το κάθε ένα, κάθε τμήμα θα υλοποιείται με 2 ΟΚΜ (βλέπε σχήμα 5.13.2).



Σχήμα 5.13.2 Τμήμα T_i , με $i=0, 1, 2, 3$.

Άσκηση 5.14

Έχετε κύρια μνήμη με οργάνωση μιας ψηφιολέξης (byte) ανά θέση μνήμης. Για την προσπέλαση της κρυφής μνήμης θεωρήστε ότι η διεύθυνση αποτελείται από τρία πεδία:

13 δυαδικά ψηφία	14 δυαδικά ψηφία	5 δυαδικά ψηφία
------------------	------------------	-----------------

Το μέγεθος της κρυφής μνήμης είναι μικρότερο ή ίσο με 2 Mψηφιολέξεις (2 Mbytes).

α. Ποιές είναι οι πιθανές οργανώσεις της κρυφής μνήμης και ποια είναι η χωρητικότητά της σε κάθε περίπτωση;

β. Να υπολογίσετε σε κάθε περίπτωση ποιος είναι ο αριθμός των μπλοκ της κύριας μνήμης με χωρητικότητα 2^{32} ψηφιολέξεις που μπορούν να μεταφερθούν (όχι ταυτόχρονα φυσικά) στο ίδιο πλαίσιο ή σύνολο, ανάλογα της οργάνωσης, της κρυφής μνήμης.

Απάντηση

α. Η οργάνωση πλήρους συσχέτισης αποκλείεται διότι σ' αυτή την περίπτωση για την προσπέλαση της κρυφής μνήμης θεωρούμε ότι η διεύθυνση αποτελείται από δύο πεδία. Επομένως έχουμε ή οργάνωση μονοσήμαντης απεικόνισης ή οργάνωση τ-τρόπων συνόλου συσχέτισης. Και στις δύο περιπτώσεις τα 5 λιγότερο σημαντικά δυαδικά ψηφία δηλώνουν τη διεύθυνση της ψηφιολέξης μέσα στο πλαίσιο της κρυφής μνήμης και στο μπλοκ της κύριας μνήμης. Επομένως, το μέγεθος του πλαισίου και του μπλοκ είναι 2^5 ψηφιολέξεις. Επίσης και στις δύο περιπτώσεις τα 13 περισσότερα σημαντικά δυαδικά ψηφία δηλώνουν την ετικέτα.

i. Οργάνωση μονοσήμαντης απεικόνισης.

Τότε τα 14 δυαδικά ψηφία δηλώνουν τη διεύθυνση του πλαισίου, επομένως έχουμε 2^{14} πλαίσια. Αφού το μέγεθος του πλαισίου είναι 2^5 ψηφιολέξεις η χωρητικότητα της κρυφής μνήμης θα είναι $2^{14} \times 2^5$ ψηφιολέξεις = 2^{19} ψηφιολέξεις = 512 Kψηφιολέξεις.

ii. Οργάνωση 2-τρόπων συνόλου συσχέτισης.

Τότε τα 14 δυαδικά ψηφία δηλώνουν τη διεύθυνση του συνόλου, επομένως έχουμε 2^{14} σύνολα. Αφού η οργάνωση είναι 2-τρόπων συνόλου συσχέτισης θα έχουμε 2 πλαίσια ανά σύνολο, άρα $2^{14} \times 2$ πλαίσια = 2^{15} πλαίσια. Το μέγεθος του πλαισίου είναι 2^5 ψηφιολέξεις. Επομένως η χωρητικότητα της κρυφής μνήμης θα είναι $2^{15} \times 2^5$ ψηφιολέξεις = 2^{20} ψηφιολέξεις = 1 Mψηφιολέξεις.

iii. Οργάνωση 3-τρόπων συνόλου συσχέτισης.

Τότε τα 14 δυαδικά ψηφία δηλώνουν τη διεύθυνση του συνόλου, επομένως έχουμε 2^{14} σύνολα. Αφού η οργάνωση είναι 3-τρόπων συνόλου συσχέτισης θα έχουμε 3 πλαίσια ανά σύνολο, άρα $2^{14} \times 3$ πλαίσια. Το μέγεθος του πλαισίου είναι 2^5 ψηφιολέξεις. Επομένως η χωρητικότητα της κρυφής μνήμης θα είναι $(2^{14} \times 3) \times 2^5$ ψηφιολέξεις = 1536 Kψηφιολέξεις.

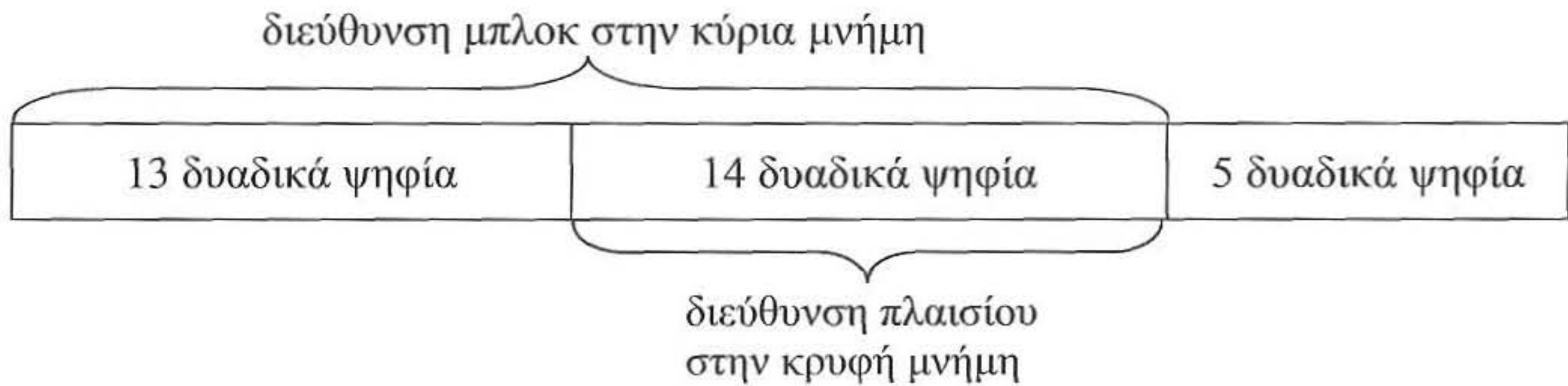
iv. Οργάνωση 4-τρόπων συνόλου συσχέτισης.

Τα 14 δυαδικά ψηφία δηλώνουν τη διεύθυνση του συνόλου, επομένως έχουμε και πάλι 2^{14} σύνολα. Όμως αφού η οργάνωση είναι 4-τρόπων συνόλου συσχέτισης θα έχουμε 4 πλαίσια ανά σύνολο, άρα $2^{14} \times 4$ πλαίσια = 2^{16} πλαίσια. Το μέγεθος του πλαισίου είναι 2^5 ψηφιολέξεις. Επομένως η χωρητικότητα της κρυφής μνήμης θα είναι $2^{16} \times 2^5$ ψηφιολέξεις = 2^{21} ψηφιολέξεις = 2 Mψηφιολέξεις.

Παρατηρούμε ότι αυξάνοντας τον αριθμό των πλαισίων ανά σύνολο, αυξάνεται η χωρητικότητα της κρυφής μνήμης. Επομένως, αν αυξήσουμε περαιτέρω το πλήθος των πλαισίων ανά σύνολο η χωρητικότητα της κρυφής μνήμης θα ξεπεράσει τις 2 Mψηφιολέξεις.

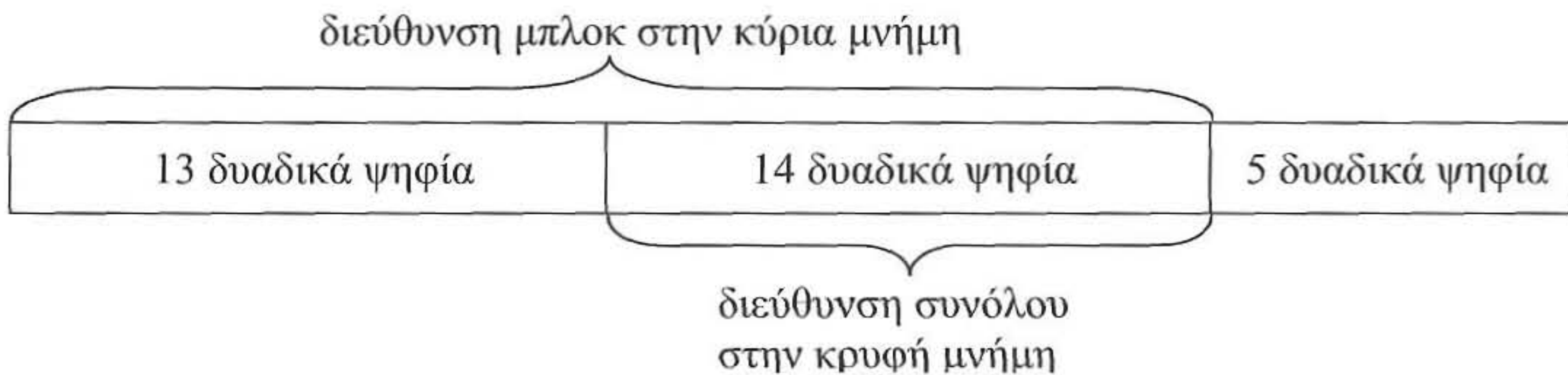
β.

i. Οργάνωση μονοσήμαντης απεικόνισης.



Κάθε μπλοκ της κύριας μνήμης μεταφέρεται σε ένα πλαίσιο της κρυφής μνήμης. Όλα τα μπλοκ της κύριας μνήμης των οποίων οι διευθύνσεις έχουν τα ίδια 14 λιγότερο σημαντικά δυαδικά ψηφία μεταφέρονται (σε διαφορετικές χρονικές περιόδους) στο ίδιο πλαίσιο της κρυφής μνήμης. Το πλήθος αυτών των μπλοκ είναι 2^{13} , επομένως στο ίδιο πλαίσιο της κρυφής μνήμης μπορούν να μεταφερθούν 2^{13} μπλοκ.

ii. Οργάνωση 2-τρόπων συνόλου συσχέτισης.



Κάθε μπλοκ της κύριας μνήμης μεταφέρεται σε ένα από τα δύο πλαίσια ενός συνόλου της κρυφής μνήμης. Όλα τα μπλοκ που έχουν τα ίδια 14 λιγότερο σημαντικά δυαδικά ψηφία μεταφέρονται (σε διαφορετικές χρονικές περιόδους) σε ένα από τα δύο πλαίσια του ίδιου συνόλου της κρυφής μνήμης. Το πλήθος αυτών των μπλοκ είναι 2^{13} , επομένως στο ίδιο σύνολο της κρυφής μνήμης μπορούν να μεταφερθούν 2^{13} μπλοκ.

Στην περίπτωση οργάνωσης 3-τρόπων και 4-τρόπων συνόλου συσχέτισης, για τον ίδιο λόγο, στο ίδιο σύνολο της κρυφής μνήμης μπορούν να μεταφερθούν 2^{13} μπλοκ.

Άσκηση 5.15

Θεωρήστε υπολογιστή με κύρια μνήμη 32 M ψηφιολέξεις και οργάνωση μιας ψηφιολέξης ανά θέση μνήμης. Η ιδεατή μνήμη χρησιμοποιεί λογικές διευθύνσεις των 39 δυαδικών ψηφίων και μέγεθος σελίδας 4 Kψηφιολέξεις. Υποθέστε ότι η ιδεατή μνήμη υλοποιείται με την τεχνική της σελιδοποίησης. Ο Πίνακας σελίδων υλοποιείται σε τρία επίπεδα υπό μορφή δένδρου και οι υποπίνακες του πίνακα σελίδων αποθηκεύονται στην ιδεατή μνήμη. Για την προσπέλαση οποιουδήποτε υποπίνακα του πίνακα σελίδων χρησιμοποιείται το ίδιο πλήθος δυαδικών ψηφίων. Θεωρήστε ότι οι πρώτες λογικές διευθύνσεις που παράγονται από τον επεξεργαστή είναι οι ακόλουθες $7235672237753_{(8)}$, $7235672237754_{(8)}$, $7235672264340_{(8)}$, $7235632214354_{(8)}$ και $7721232216754_{(8)}$. Να σχεδιάσετε τη δομή του πίνακα σελίδων που βρίσκεται στην κύρια μνήμη μετά από καθεμία λογική διεύθυνση που παράγει ο επεξεργαστής. Να δικαιολογήσετε τις απαντήσεις σας.

Απάντηση

Η χωρητικότητα της κυρίας μνήμης είναι 32 M ψηφιολέξεις = $2^5 \times 2^{20}$ ψηφιολέξεις = 2^{25} ψηφιολέξεις με οργάνωση μιας ψηφιολέξης ανά θέση μνήμης, επομένως η φυσική διεύθυνση είναι 25 δυαδικών ψηφίων.

Εφόσον το μέγεθος της σελίδας είναι 4 Kψηφιολέξεις = $2^2 \times 2^{10}$ ψηφιολέξεις = 2^{12} ψηφιολέξεις, τα 12 λιγότερο σημαντικά δυαδικά ψηφία της φυσικής διεύθυνσης αποτελούν τη διεύθυνση μέσα στη σελίδα, δμΣ και τα υπόλοιπα 13 πιο σημαντικά δυαδικά ψηφία αποτελούν τη φυσική διεύθυνση της σελίδας, φδΣ.

Επίσης από τα 39 δυαδικά ψηφία της λογικής διεύθυνσης τα 12 λιγότερο σημαντικά δυαδικά ψηφία αποτελούν τη διεύθυνση μέσα στη σελίδα, δμΣ, και τα υπόλοιπα $39-12=27$ αποτελούν τον αριθμό λογικής σελίδας, ΑΛΣ.

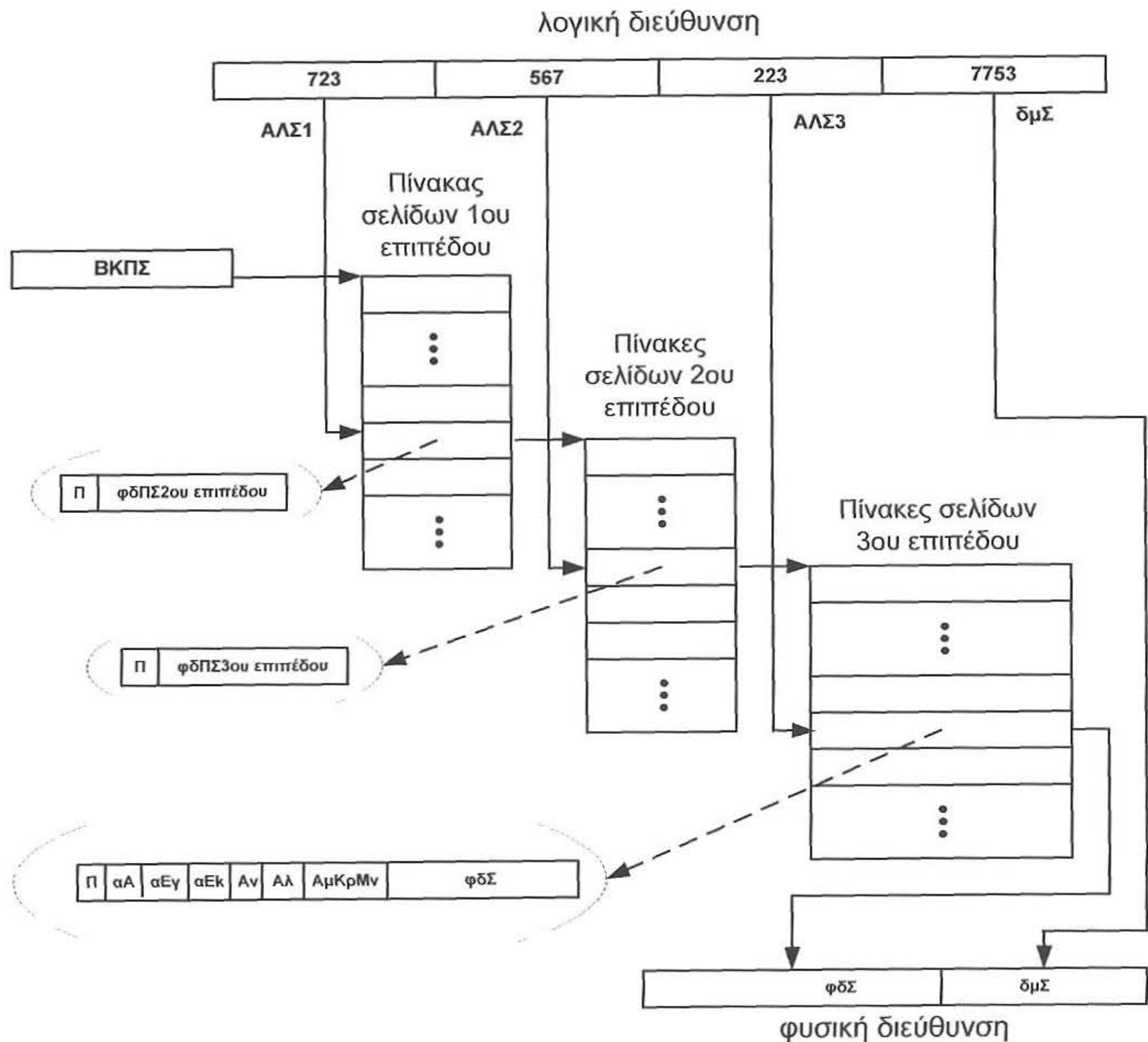
Επειδή ο πίνακας σελίδων υλοποιείται σε τρία επίπεδα υπό μορφή δένδρου, ο ΑΛΣ αποτελείται από τρία πεδία ΑΛΣ1, ΑΛΣ2 και ΑΛΣ3. Αφού για την προσπέλαση οποιουδήποτε υποπίνακα του πίνακα σελίδων χρησιμοποιείται το ίδιο πλήθος δυαδικών ψηφίων, συνεπάγεται ότι για τον ΑΛΣ1 χρησιμοποιούνται τα $27/3=9$ πιο σημαντικά δυαδικά ψηφία του ΑΛΣ, για τον ΑΛΣ2 χρησιμοποιούνται τα $27/3=9$ αμέσως λιγότερο σημαντικά δυαδικά ψηφία του ΑΛΣ και για τον ΑΛΣ3 χρησιμοποιούνται τα επόμενα $27/3=9$ λιγότερο σημαντικά δυαδικά ψηφία του ΑΛΣ.

Στο σχήμα 5.15.1 δίνεται η δομή του πίνακα σελίδων μετά την παραγωγή από τον επεξεργαστή της λογικής διεύθυνσης $7235672237753_{(8)}$ και την ενημέρωση της δομής του πίνακα σελίδων. Η δομή του πίνακα σελίδων δεν αλλάζει με τη παραγωγή από τον επεξεργαστή της λογικής διεύθυνσης $7235672237754_{(8)}$, διότι αυτή η λογική διεύθυνση ανήκει στην ίδια λογική σελίδα με την προηγούμενη λογική σελίδα.

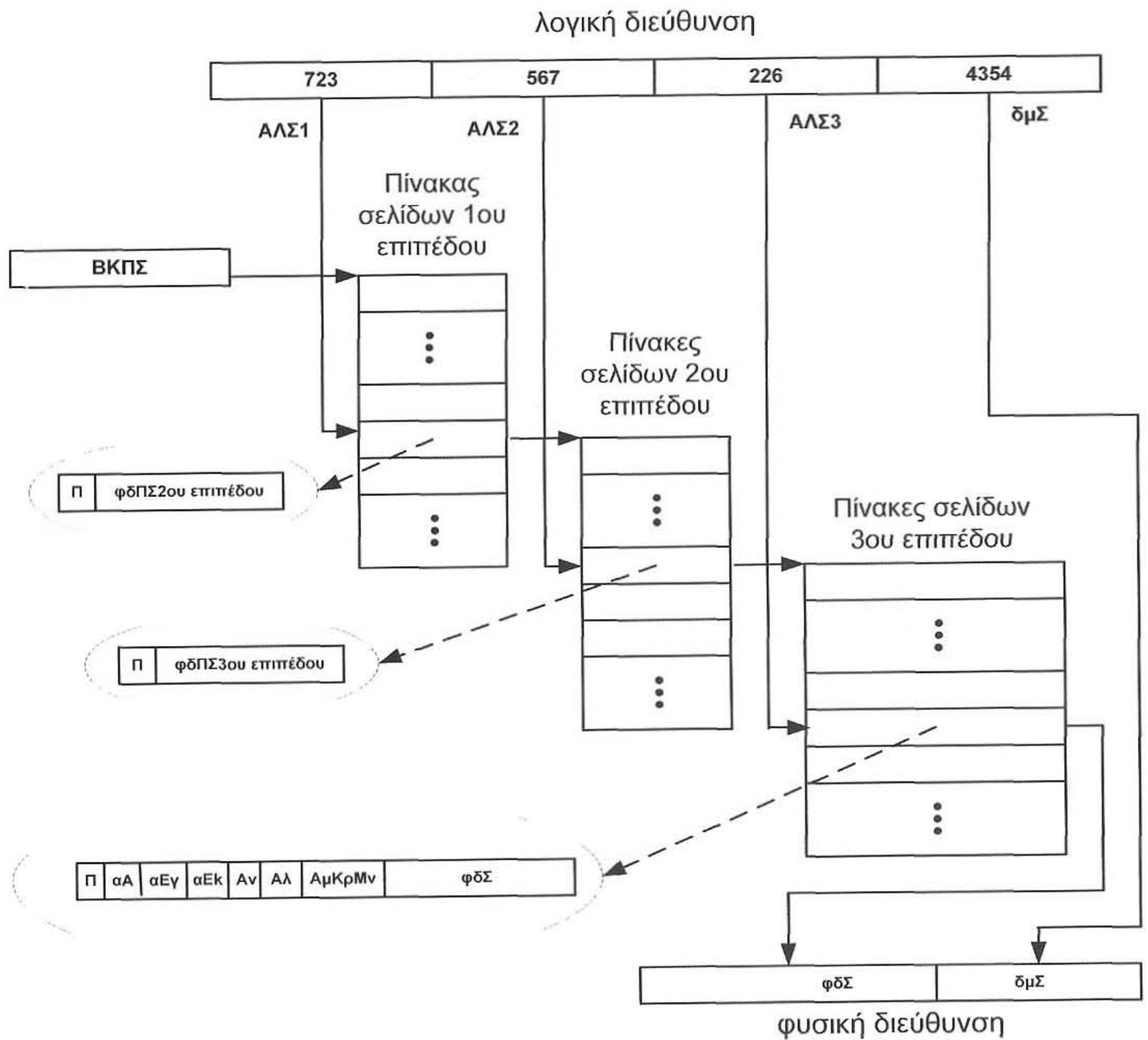
Η δομή του πίνακα σελίδων μετά τη διεύθυνση $7235672264340_{(8)}$, δίνεται από το σχήμα 5.15.2. Παρατηρούμε ότι δεν έχει προστεθεί κανένας υποπίνακας, διότι αυτή η λογική διεύθυνση διαφέρει από τις προηγούμενες μόνο στο πεδίο ΑΛΣ3.

Στο σχήμα 5.15.3 δίνεται η δομή του πίνακα σελίδων μετά τη διεύθυνση $7235632214354_{(8)}$. Επειδή η διεύθυνση αυτή δε διαφέρει από τις προηγούμενες στο πεδίο ΑΛΣ1, αλλά διαφέρει από τις προηγούμενες στο πεδίο ΑΛΣ2, σε σχέση με την προηγούμενη δομή προστίθεται ένας υποπίνακας στο τρίτο επίπεδο του δένδρου.

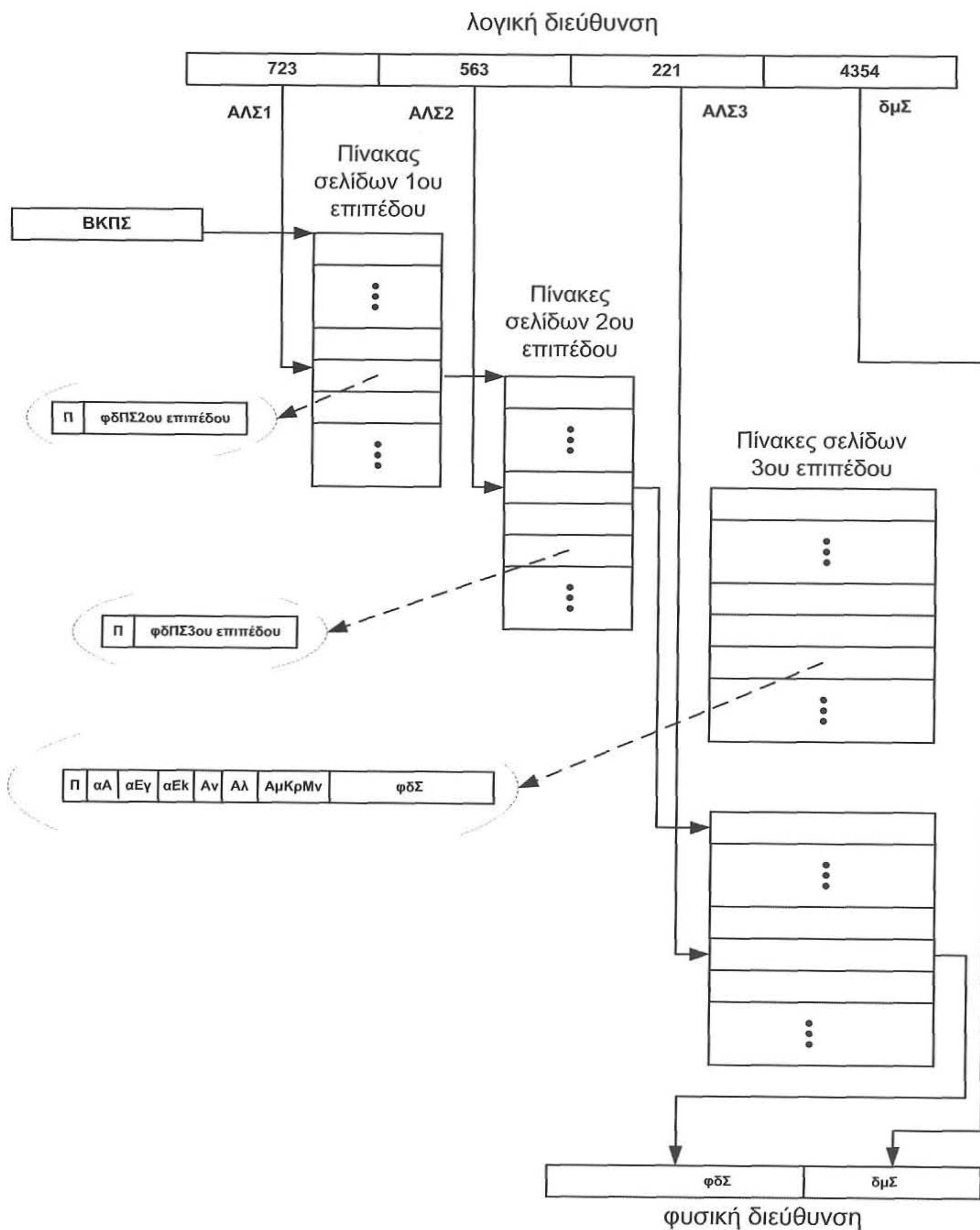
Στο σχήμα 5.15.4 δίνεται η δομή του πίνακα σελίδων μετά τη διεύθυνση $7721232216754_{(8)}$. Επειδή η διεύθυνση αυτή διαφέρει από τις προηγούμενες στο πεδίο ΑΛΣ1, σε σχέση με την προηγούμενη δομή (σχήμα 5.15.3) στη ρίζα του δένδρου προστίθεται ένα νέο κλαδί, δηλαδή ένας υποπίνακας στο δεύτερο επίπεδο και ένας στο τρίτο.



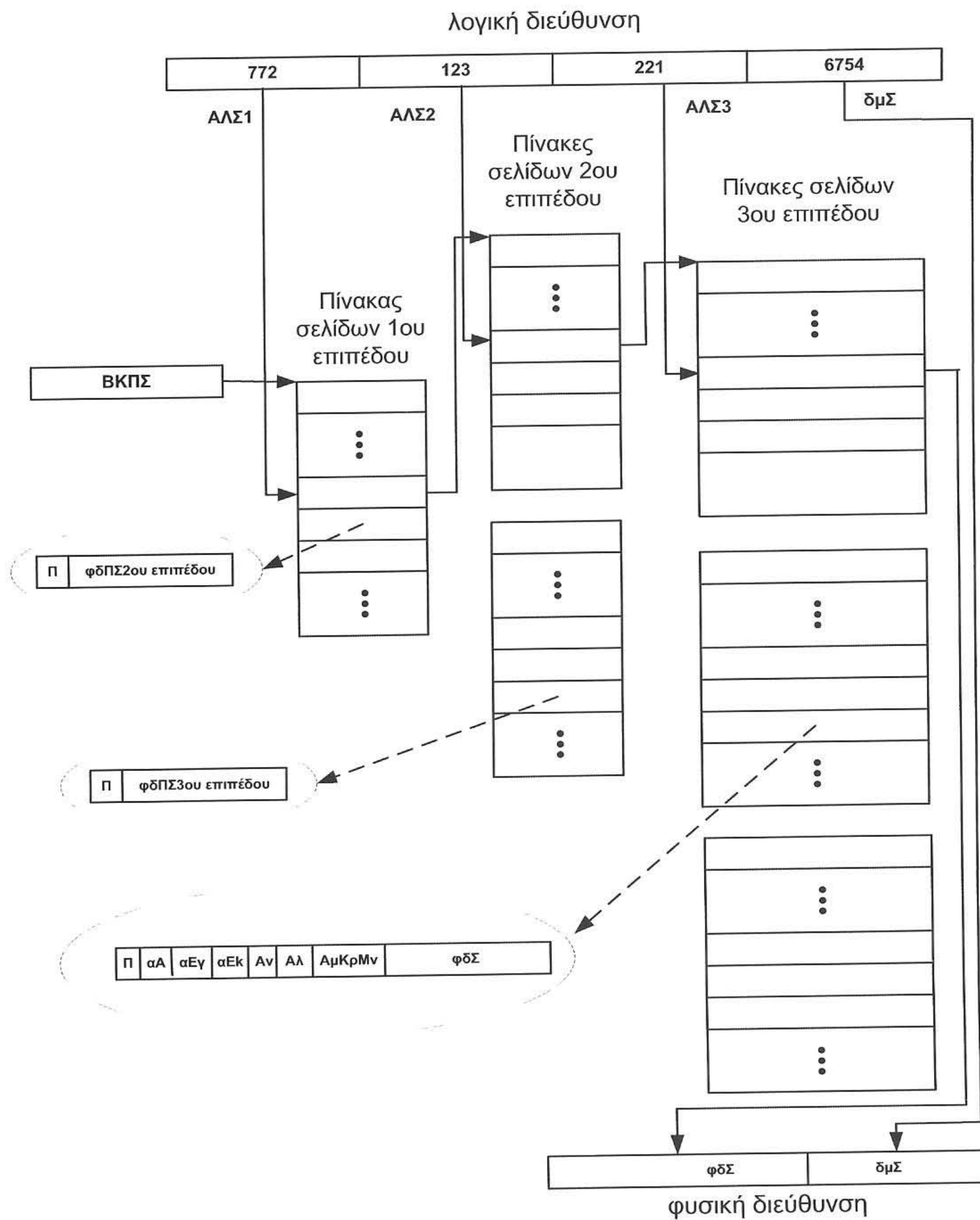
Σχήμα 5.15.1



Σχήμα 5.15.2



Σχήμα 5.15.3



Σχήμα 5.15.4

Άσκηση 5.16

Θεωρήστε υπολογιστή με κύρια μνήμη 32 Μ ψηφιολέξεις και οργάνωση μιας ψηφιολέξης ανά θέση μνήμης. Η ιδεατή μνήμη χρησιμοποιεί λογικές διευθύνσεις των 36 δυαδικών ψηφίων και μέγεθος σελίδας 4 Κψηφιολέξεις. Υποθέστε ότι η ιδεατή μνήμη υλοποιείται με την τεχνική της σελιδοποίησης. Να δικαιολογήσετε τις απαντήσεις σας.

α. Να υπολογίσετε το μέγεθος της κυρίας μνήμης που απαιτείται για την αποθήκευση του πίνακα σελίδων εάν αυτός υλοποιείται σε ένα επίπεδο.

β. Θεωρήστε μια διαδικασία στην οποία ο χώρος των λογικών διευθύνσεων καταλαμβάνει 256 σελίδες. Να υπολογίσετε το μέγεθος της κυρίας μνήμης που απαιτείται για την αποθήκευση του πίνακα σελίδων στη χειρότερη και στην καλύτερη των περιπτώσεων εάν αυτός υλοποιείται σε δύο επίπεδα.

Απάντηση

Η χωρητικότητα της κυρίας μνήμης είναι 32 Μ ψηφιολέξεις = $2^5 \times 2^{20}$ ψηφιολέξεις = 2^{25} ψηφιολέξεις με οργάνωση μιας ψηφιολέξης ανά θέση μνήμης, επομένως η φυσική διεύθυνση είναι 25 δυαδικών ψηφίων. Εφόσον το μέγεθος της σελίδας είναι 4 Κψηφιολέξεις = $2^2 \times 2^{10}$ ψηφιολέξεις = 2^{12} ψηφιολέξεις, τα 12 λιγότερο σημαντικά δυαδικά ψηφία της φυσικής διεύθυνσης αποτελούν τη διεύθυνση μέσα στη σελίδα, δμΣ και τα υπόλοιπα 13 πιο σημαντικά δυαδικά ψηφία αποτελούν τη φυσική διεύθυνση της σελίδας, φδΣ.

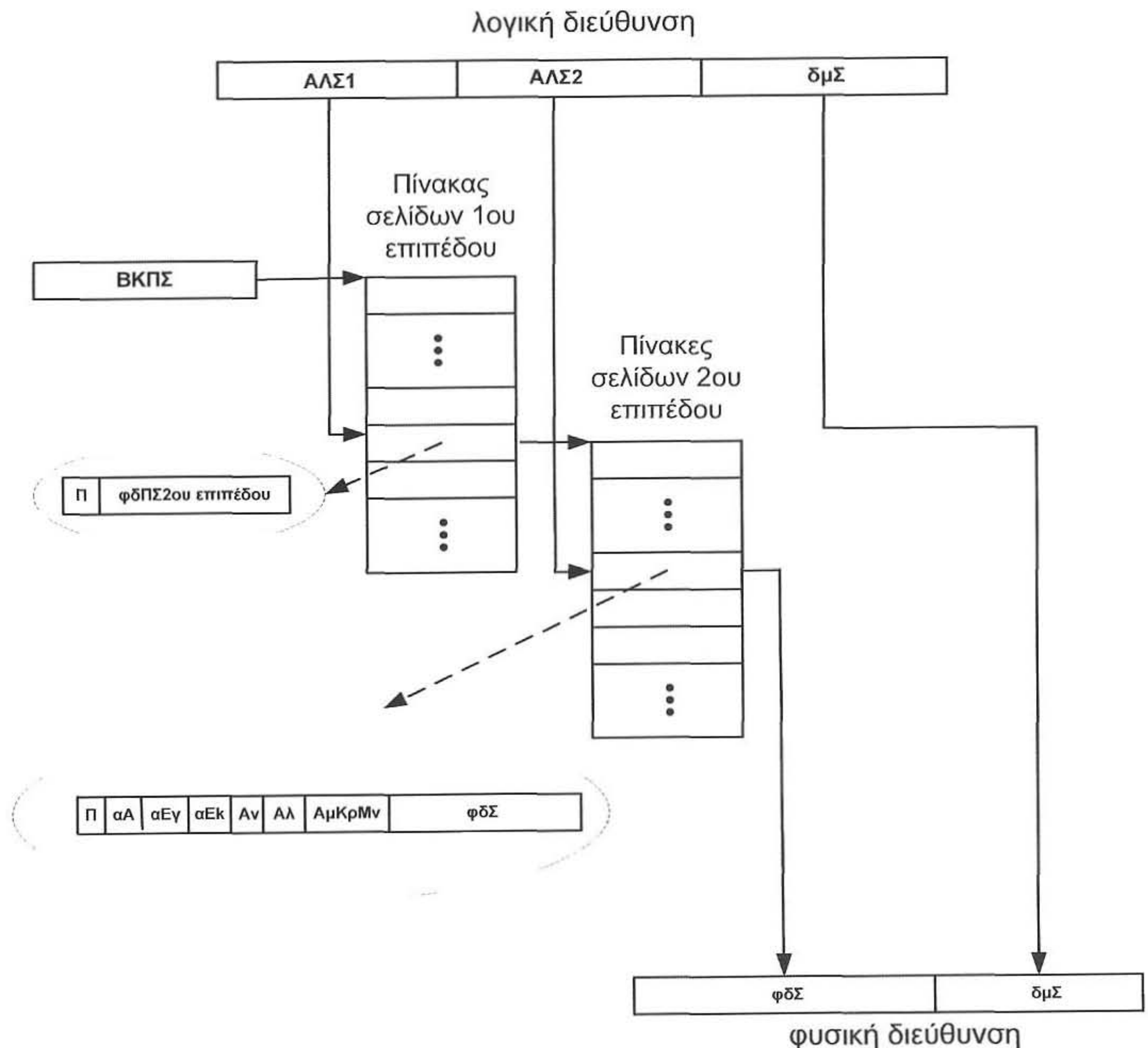
α. Υλοποίηση του πίνακα σελίδων σε ένα επίπεδο.

Εφόσον η λογική διεύθυνση είναι των 36 δυαδικών ψηφίων και τα 12 λιγότερο σημαντικά από αυτά δηλώνουν τη διεύθυνση μέσα στη σελίδα, τα υπόλοιπα 24 πιο σημαντικά δυαδικά ψηφία θα δηλώνουν τον αριθμό της λογικής σελίδας, ΑΛΣ. Επομένως ο πίνακας σελίδων θα έχει 2^{24} θέσεις, μια θέση για κάθε λογική σελίδα. Κάθε μία θέση θα περιέχει τα δυαδικά ψηφία ελέγχου (7 δυαδικά ψηφία) και χώρο για τη φυσική διεύθυνση σελίδας (13 δυαδικά ψηφία), δηλαδή $7+13$ δυαδικά ψηφία = 20 δυαδικά ψηφία. Επομένως η αποθήκευση του πίνακα σελίδων απαιτεί $2^{24} \times (7+13)$ δυαδικά ψηφία = $2^{21} \times (7+13)$ ψηφιολέξεις = 40 Μψηφιολέξεις. Παρατηρούμε ότι δεν χωράει στην κύρια μνήμη.

β. Υλοποίηση του πίνακα σελίδων σε δύο επίπεδα.

Σε αυτή την περίπτωση η λογική διεύθυνση θεωρούμε ότι αποτελείται από τρία πεδία. Τα 12 λιγότερο σημαντικά δυαδικά ψηφία δηλώνουν και πάλι τη διεύθυνση μέσα στη σελίδα. Υπάρχουν διάφοροι τρόποι να μοιράσουμε τα υπόλοιπα 24 πιο σημαντικά δυαδικά ψηφία σε δύο μέρη. Θα εξετάσουμε τρεις περιπτώσεις.

β.1 Θεωρούμε ότι τα 12 πιο σημαντικά δυαδικά ψηφία του ΑΛΣ χρησιμοποιούνται ως δείκτης για τον υποπίνακα του πρώτου επιπέδου και τα 12 λιγότερο σημαντικά δυαδικά ψηφία χρησιμοποιούνται ως δείκτης για τον υποπίνακα του δεύτερου επιπέδου (βλέπε σχήμα 5.16.1, με τα πεδία ΑΛΣ1, ΑΛΣ2 και δμΣ κάθε ένα των 12 δυαδικών ψηφίων).



Σχήμα 5.16.1

Στην καλύτερη περίπτωση οι διευθύνσεις των 256 σελίδων έχουν τα ίδια 12 πιο σημαντικά δυαδικά ψηφία. Τότε ο πίνακας σελίδων αποτελείται από τον υποπίνακα του πρώτου επιπέδου και ένα υποπίνακα του δεύτερου επιπέδου. Επομένως η αποθήκευση του πίνακα σελίδων απαιτεί $2^{12} \times (1+13)$ δυαδικά ψηφία + $2^{12} \times (7+13)$ δυαδικά ψηφία = $2^{12} \times 34$ δυαδικά ψηφία = $2^9 \times 34$ ψηφιολέξεις = 17.408 ψηφιολέξεις.

Στη χειρότερη περίπτωση τα 12 πιο σημαντικά δυαδικά ψηφία των διευθύνσεων και των 256 σελίδων έχουν διαφορετικές τιμές. Τότε ο πίνακας σελίδων αποτελείται από τον υποπίνακα του πρώτου επιπέδου και 256 υποπίνακες του δεύτερου επιπέδου. Επομένως η αποθήκευση του πίνακα σελίδων απαιτεί $2^{12} \times (1+13)$ δυαδικά ψηφία + $256 \times 2^{12} \times (7+13)$ δυαδικά ψηφία = $2^{12} \times (14+256 \times 20)$ δυαδικά ψηφία = $2^{12} \times 5.134$ δυαδικά ψηφία = $2^{10} \times 2.567$ ψηφιολέξεις = 2.567 Kψηφιολέξεις.

β.2 Θεωρούμε ότι τα 14 πιο σημαντικά δυαδικά ψηφία χρησιμοποιούνται ως δείκτης για τον υποπίνακα του πρώτου επιπέδου και τα 10 λιγότερο σημαντικά δυαδικά ψηφία χρησιμοποιούνται ως δείκτης για τον υποπίνακα του δεύτερου επιπέδου (βλέπε σχήμα 5.16.1, με ΑΛΣ1 των 14 δυαδικών ψηφίων, ΑΛΣ2 των 10 δυαδικών ψηφίων και δμΣ των 12 δυαδικών ψηφίων).

Στην καλύτερη περίπτωση οι διευθύνσεις των 256 σελίδων έχουν τα ίδια 14 πιο σημαντικά δυαδικά ψηφία. Τότε ο πίνακας σελίδων αποτελείται από τον υποπίνακα του πρώτου επιπέδου και ένα υποπίνακα του δεύτερου επιπέδου. Επομένως η αποθήκευση του πίνακα σελίδων απαιτεί $2^{14} \times (1+13)$ δυαδικά ψηφία + $2^{10} \times (7+13)$ δυαδικά ψηφία = $2^{10} \times (2^4 \times 14 + 20)$ δυαδικά ψηφία = 31.232 ψηφιολέξεις.

Στη χειρότερη περίπτωση τα 14 πιο σημαντικά δυαδικά ψηφία των διευθύνσεων και των 256 σελίδων έχουν διαφορετικές τιμές. Τότε ο πίνακας σελίδων αποτελείται από τον υποπίνακα του πρώτου επιπέδου και 256 υποπίνακες του δεύτερου επιπέδου. Επομένως η αποθήκευση του πίνακα σελίδων απαιτεί $2^{14} \times (1+13)$ δυαδικά ψηφία + $256 \times 2^{10} \times (7+13)$ δυαδικά ψηφία = $2^{10} \times (2^4 \times 14 + 256 \times 20)$ δυαδικά ψηφία = $2^{10} \times 5.344$ δυαδικά ψηφία = 668 Kψηφιολέξεις.

β.3 Θεωρούμε ότι τα 10 πιο σημαντικά δυαδικά ψηφία χρησιμοποιούνται ως δείκτης για τον υποπίνακα του πρώτου επιπέδου και τα 14 λιγότερο σημαντικά δυαδικά ψηφία χρησιμοποιούνται ως δείκτης για τον υποπίνακα του δεύτερου επιπέδου (βλέπε σχήμα 5.16.1, με ΑΛΣ1 των 10 δυαδικών ψηφίων, ΑΛΣ2 των 14 δυαδικών ψηφίων και δμΣ των 12 δυαδικών ψηφίων).

Στην καλύτερη περίπτωση οι διευθύνσεις των 256 σελίδων έχουν τα ίδια 10 πιο σημαντικά δυαδικά ψηφία. Τότε ο πίνακας σελίδων αποτελείται από τον υποπίνακα του πρώτου επιπέδου και ένα υποπίνακα του δεύτερου επιπέδου. Επομένως η αποθήκευση του πίνακα σελίδων απαιτεί $2^{10} \times (1+13)$ δυαδικά ψηφία + $2^{14} \times (7+13)$ δυαδικά ψηφία = $2^{10} \times (14 + 2^4 \times 20)$ δυαδικά ψηφία = $2^{10} \times 334$ δυαδικά ψηφία = $2^7 \times 334$ ψηφιολέξεις = 42.752 ψηφιολέξεις = 41,75 Kψηφιολέξεις.

Στη χειρότερη περίπτωση τα 10 πιο σημαντικά δυαδικά ψηφία των διευθύνσεων και των 256 σελίδων έχουν διαφορετικές τιμές. Τότε ο πίνακας σελίδων αποτελείται από τον υποπίνακα του πρώτου επιπέδου και 256 υποπίνακες του δεύτερου επιπέδου. Επομένως η αποθήκευση του πίνακα σελίδων απαιτεί $2^{10} \times (1+13)$ δυαδικά ψηφία + $256 \times 2^{14} \times (7+13)$ δυαδικά ψηφία = $2^{10} \times (14 + 256 \times 2^4 \times 20)$ δυαδικά ψηφία = $2^{10} \times 81.934$ δυαδικά ψηφία = 81.934 Kδυαδικά ψηφία = 10.241,75 Kψηφιολέξεις.

Άσκηση 5.17 (5.11 βιβλίου)

Θεωρήστε υπολογιστή με κύρια μνήμη χωρητικότητας 16 Mψηφιολέξεις (16 Mbytes) και οργάνωση μίας ψηφιολέξης (byte) ανά θέση μνήμης, ιδεατή μνήμη που χρησιμοποιεί λογικές διευθύνσεις των 36 δυαδικών ψηφίων και μέγεθος σελίδας 4 Kψηφιολέξεις.

α. Υποθέστε ότι η ιδεατή μνήμη υλοποιείται με την τεχνική της σελιδοποίησης με πίνακα σελίδων ενός επιπέδου. Να υπολογίσετε το μέγεθος της κύριας μνήμης που απαιτείται για την αποθήκευση του πίνακα σελίδων.

β. Υποθέστε ότι η ιδεατή μνήμη υλοποιείται με την τεχνική του αντιστραμμένου πίνακα σελίδων (inverted page table) με δύο αντιστοιχίσεις ανά θέση του πίνακα. Η σχετική διεύθυνση μέσα στον αντιστραμμένο πίνακα σελίδων, $\sigma\delta\mu\Lambda\Pi\Sigma_i$, δίνεται από τη συνάρτηση διασποράς (hash function) $\sigma\delta\mu\Lambda\Pi\Sigma_i = \Lambda\Lambda\Sigma_i + \Lambda\Lambda\Sigma_{i+4} \bmod 8$, για i από 1 έως και 4. Το $\Lambda\Lambda\Sigma_i$ είναι το i ψηφίο, στο οκταδικό, του Αριθμού Λογικής Σελίδας, αρχίζοντας τη μέτρηση από το λιγότερο σημαντικό οκταδικό ψηφίο.

i. Να υπολογίσετε το μέγεθος της κύριας μνήμης που καταλαμβάνει ο αντιστραμμένος πίνακας σελίδων.

ii. Θεωρήστε ότι αρχικά ο αντιστραμμένος πίνακας σελίδων είναι κενός και δώστε το περιεχόμενό του μετά την παραγωγή από τον επεξεργαστή της επόμενης ακολουθίας λογικών διευθύνσεων στο οκταδικό: 156032010000, 370667171234, 043732174552 και 320115600022. Θεωρήστε ότι αυτές οι λογικές διευθύνσεις, μετά την προσκόμιση των απαιτούμενων σελίδων στην κύρια μνήμη, αντιστοιχούν στις φυσικές διευθύνσεις: 17620000, 63551234, 46004552 και 06650022.

Θεωρήστε ότι το περιεχόμενο του βασικού καταχωρητή πίνακα σελίδων είναι μηδέν.

Απάντηση

α. Λαμβάνοντας υπόψη ότι έχουμε λογικές διευθύνσεις των 36 δυαδικών ψηφίων, η κύρια μνήμη έχει οργάνωση μίας ψηφιολέξης (byte) ανά θέση μνήμης και το μέγεθος σελίδας είναι $4 \text{ Kψηφιολέξεις} = 2^2 \times 2^{10} \text{ ψηφιολέξεις} = 2^{12} \text{ ψηφιολέξεις}$, συμπεραίνουμε ότι ο χώρος λογικών διευθύνσεων αποτελείται από $2^{36}/2^{12} = 2^{24}$ σελίδες. Επομένως ο ΑΛΣ αποτελείται από 24 δυαδικά ψηφία και ο πίνακας σελίδων θα έχει 2^{24} θέσεις. Η κύρια μνήμη αποτελείται από $16 \text{ Mψηφιολέξεις} / 4 \text{ Kψηφιολέξεις} = (2^4 \times 2^{20}) / (2^2 \times 2^{10}) = 2^{24} / 2^{12} = 2^{12}$ πλαίσια σελίδων, άρα η φυσική διεύθυνση σελίδας αποτελείται από 12 δυαδικά ψηφία. Κάθε θέση του πίνακα σελίδων θα έχει 7 δυαδικά ψηφία ελέγχου και 12 δυαδικά ψηφία για τη φυσική διεύθυνση σελίδας, δηλαδή $7+12=19$ δυαδικά ψηφία συνολικά. Ο συνολικός χώρος της κύριας μνήμης που απαιτείται για την αποθήκευση του πίνακα σελίδων είναι 19×2^{24} δυαδικά ψηφία $= 19 \times 2^{21}$ ψηφιολέξεις $= 38 \text{ Mψηφιολέξεις}$.

β.i. Όπως υπολογίσαμε στο προηγούμενο ερώτημα, η κύρια μνήμη αποτελείται από 2^{12} πλαίσια σελίδων. Ο αντιστραμμένος πίνακας σελίδων έχει τόσες θέσεις όσα είναι τα πλαίσια σελίδων της κυρίας μνήμης, επομένως θα αποτελείται από 2^{12} θέσεις. Ο χώρος λογικών διευθύνσεων αποτελείται από $2^{36}/2^{12} = 2^{24}$ σελίδες, επομένως ο ΑΛΣ αποτελείται από 24 δυαδικά ψηφία. Στην περίπτωση του αντιστραμμένου πίνακα σελίδων, για κάθε αντιστοίχιση απαιτούνται $7+24+12$ δυαδικά ψηφία. Κάθε θέση του αντιστραμμένου πίνακα σελίδων περιέχει πληροφορία για δύο αντιστοιχίσεις, άρα απαιτούνται $2 \times (7+24+12) = 86$ δυαδικά ψηφία ανά θέση. Ο συνολικός χώρος της κύριας μνήμης που καταλαμβάνεται από τον αντιστραμμένο πίνακα σελίδων είναι 86×2^{12} δυαδικά ψηφία, δηλαδή $43 \times 2^3 \times 2^{10}$ δυαδικά ψηφία ή 43 Kψηφιολέξεις.

β.ii. Λαμβάνοντας υπόψη ότι το περιεχόμενο του βασικού καταχωρητή πίνακα σελίδων είναι μηδέν, συμπεραίνουμε ότι εφαρμόζοντας τη δοθείσα συνάρτηση διασποράς στο τμήμα ΑΛΣ κάθε διεύθυνσης που παράγει ο επεξεργαστής, λαμβάνουμε τη φυσική διεύθυνση της θέσης του αντιστραμμένου πίνακα σελίδων η οποία πιθανόν να περιέχει τη ζητούμενη αντιστοίχιση. Αφού το μέγεθος κάθε σελίδας είναι $4 \text{ Kψηφιολέξεις} = 2^{12}$ ψηφιολέξεις, τα 12 λιγότερο σημαντικά δυαδικά ψηφία, ή ισοδύναμα τα 4 λιγότερο σημαντικά οκταδικά ψηφία, της λογικής διεύθυνσης αποτελούν τη διεύθυνση μέσα στη λογική σελίδα και τα υπόλοιπα 24 δυαδικά ψηφία ή ισοδύναμα τα 8 περισσότερα σημαντικά οκταδικά ψηφία, αποτελούν τον αριθμό λογικής σελίδας, ΑΛΣ.

Από την πρώτη λογική διεύθυνση 156032010000 παίρνουμε ΑΛΣ=15603201. Οπότε εφαρμόζοντας τη συνάρτηση διασποράς $\text{σδμΑΠΣ}_i = \text{ΑΛΣ}_i + \text{ΑΛΣ}_{i+4} \bmod 8$, για i από 1 έως και 4 λαμβάνουμε: $(1+0) \bmod 8 = 1$, $(0+6) \bmod 8 = 6$, $(2+5) \bmod 8 = 7$ και $(3+1) \bmod 8 = 4$. Επομένως στον αντιστραμμένο πίνακα σελίδων θα διαβαστεί το περιεχόμενο της θέσης μνήμης με φυσική διεύθυνση 4761. Θεωρήσαμε αρχικά ότι ο αντιστραμμένος πίνακας σελίδων είναι κενός. Επομένως το δυαδικό ψηφίο παρουσίας Π καθεμιάς εκ των

δύο αντιστοιχήσεων θα έχει την τιμή 0. Τότε το λειτουργικό σύστημα θα φέρει από το δίσκο τη σελίδα με ΑΛΣ =15603201 και θα την τοποθετήσει, σύμφωνα με την άσκηση, στο πλαίσιο σελίδας με φυσική διεύθυνση φδΣ1=1762. Στη συνέχεια το λειτουργικό σύστημα ενημερώνει τον αντιστραμμένο πίνακα σελίδων. Συγκεκριμένα στη θέση με διεύθυνση 4761 θέτει το πεδίο παρουσίας της μιας αντιστοίχισης στην τιμή ένα και στα πεδία ΑΛΣ1 και φδΣ1 αποθηκεύει τις τιμές 15603201 και 1762 αντίστοιχα (βλ. Πίνακα 5.17.1).

Πίνακας 5.17.1

Αντιστραμμένος Πίνακας σελίδων
(δίνουμε τα περιεχόμενα μόνο των πεδίων Π, ΑΛΣ και φδΣ)

φυσική διεύθυνση	Π	ΑΛΣ1	φδΣ1	Π	ΑΛΣ2	φδΣ2
...	0	0
4761	1	15603201	1762	0
...	0	0

Με τον ίδιο τρόπο μπορούμε να βρούμε την έξοδο της συνάρτησης διασποράς για καθεμία από τις υπόλοιπες λογικές διευθύνσεις που μας δίνει η άσκηση. Τα συνολικά αποτελέσματα δίνονται στον Πίνακα 5.17.2.

Πίνακας 5.17.2

ΑΛΣ	Έξοδος της συνάρτησης διασποράς, δηλαδή φυσική διεύθυνση θέσης στον αντιστραμμένο πίνακα σελίδων
15603201	4761
37066717	1615
04373217	3646
32011560	4761

Μπορούμε εύκολα να δούμε ότι οι σελίδες στις οποίες αντιστοιχούν και οι επόμενες τρεις λογικές διευθύνσεις που παράγει ο επεξεργαστής δεν βρίσκονται στην κύρια μνήμη. Επομένως, κάθε φορά θα προσκομίζεται μια σελίδα από το δίσκο στην κύρια μνήμη και θα ενημερώνεται ο αντιστραμμένος πίνακα σελίδων. Στους πίνακες 5.17.3, 5.17.4 και 5.17.5 δίνεται το περιεχόμενο του αντιστραμμένου πίνακα σελίδων μετά την προσκόμιση κάθε μιας σελίδας.

Πίνακας 5.17.3

φυσική διεύθυνση	Π	ΑΛΣ1	φδΣ1	Π	ΑΛΣ2	φδΣ2
...	0	0
1615	1	37066717	6355	0
...	0	0
4761	1	15603201	1762	0

Πίνακας 5.17.4

φυσική διεύθυνση	Π	ΑΛΣ1	φδΣ1	Π	ΑΛΣ2	φδΣ2
...	0	0
1615	1	37066717	6355	0
...	0	0
3646	1	04373217	4600	0
...	0	0
4761	1	15603201	1762	0

Πίνακας 5.17.5

φυσική διεύθυνση	Π	ΑΛΣ1	φδΣ1	Π	ΑΛΣ2	φδΣ2
...	0	0
1615	1	37066717	6355	0
...	0	0
3646	1	04373217	4600	0
...	0	0
4761	1	15603201	1762	1	32011560	0665

Άσκηση 5.18 (5.12 βιβλίου)

Θεωρούμε υπολογιστή με ιδεατή μνήμη (virtual memory) που υλοποιείται με την τεχνική της σελιδοποίησης. Θεωρούμε ότι υπάρχει κρυφή μνήμη πίνακα σελίδων (TLB) των 8 θέσεων με οργάνωση πλήρους συσχέτισης και κρυφή μνήμη επεξεργαστή (CPU cache memory) οργάνωσης μονοσήμαντης απεικόνισης με 16 ψηφιολέξεις ανά πλαίσιο και χωρητικότητα 32 Kψηφιολέξεις (Kbytes) η οποία προσπελάζεται με λογικές διευθύνσεις. Η λογική διεύθυνση είναι των 52 δυαδικών ψηφίων, ενώ η φυσική διεύθυνση είναι των 36 δυαδικών ψηφίων και το μέγεθος της σελίδας είναι ίσο με 4 Kψηφιολέξεις (Kbytes). Η κύρια μνήμη έχει οργάνωση μίας ψηφιολέξης ανά θέση.

Να σχεδιάσετε το κατάλληλο σχήμα που να δείχνει πως χρησιμοποιείται η λογική διεύθυνση για την προσπέλαση της κρυφής μνήμης πίνακα σελίδων και επεξεργαστή.

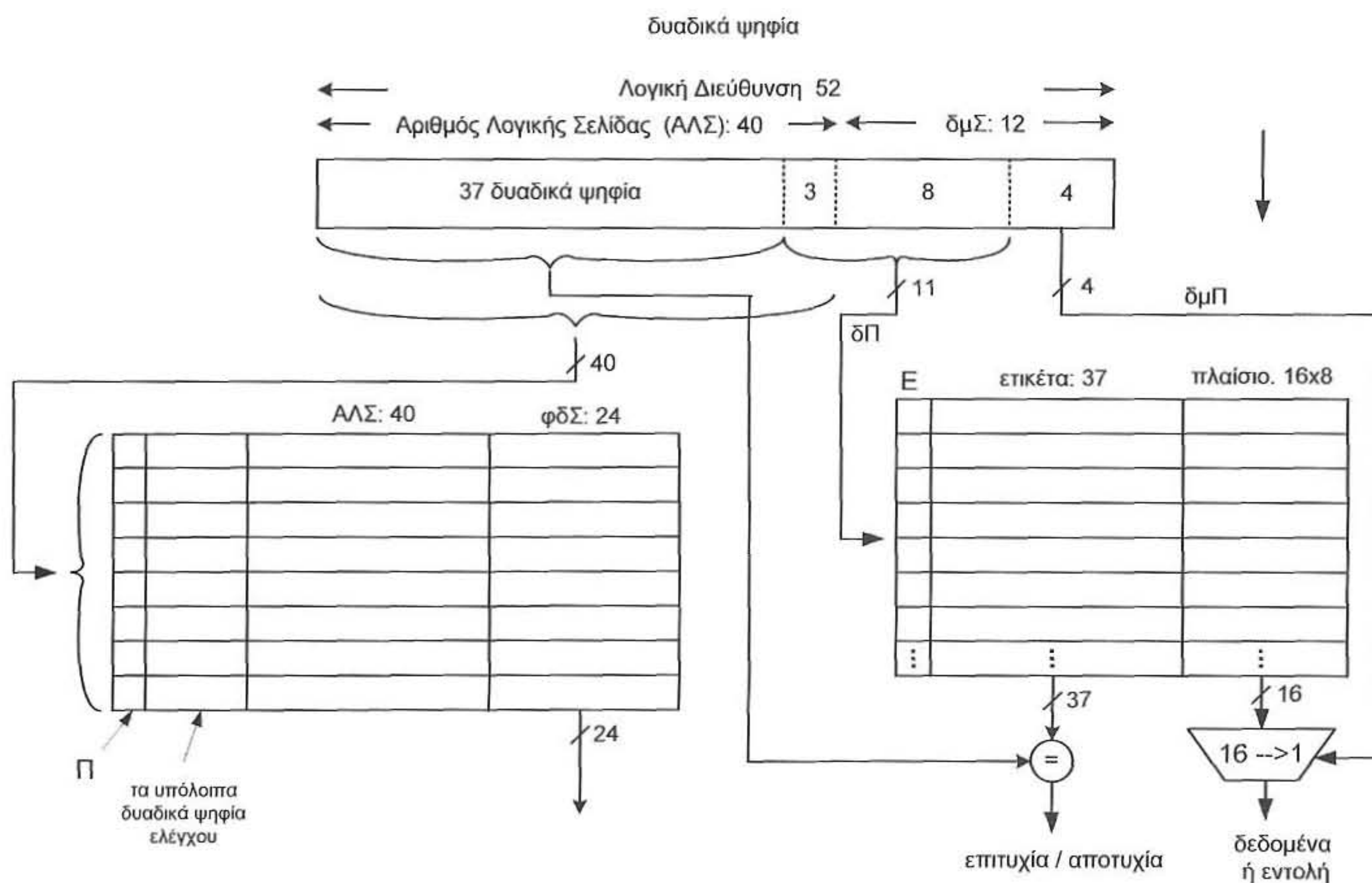
Απάντηση

Λαμβάνοντας υπόψη ότι το μέγεθος της σελίδας είναι 4 Kψηφιολέξεις = 2^{12} ψηφιολέξεις, συμπεραίνουμε ότι τα 12 λιγότερο σημαντικά δυαδικά ψηφία της λογικής διεύθυνσης καθορίζουν τη διεύθυνση μέσα στη σελίδα. Αφού η λογική διεύθυνση είναι των 52 δυαδικών ψηφίων και τα 12 λιγότερο σημαντικά δυαδικά ψηφία της λογικής διεύθυνσης καθορίζουν τη διεύθυνση μέσα στη σελίδα, τα υπόλοιπα 40 δυαδικά ψηφία αποτελούν τον Αριθμό Λογικής Σελίδας, ΑΛΣ, και χρησιμοποιούνται ως κλειδί για την προσπέλαση της κρυφής μνήμης πίνακα σελίδων (TLB).

Δεδομένου ότι η φυσική διεύθυνση είναι των 36 δυαδικών ψηφίων και το μέγεθος της σελίδας είναι ίσο με 4 Kψηφιολέξεις = 2^{12} ψηφιολέξεις, συμπεραίνουμε ότι τα 12 λιγότερο σημαντικά δυαδικά ψηφία της φυσικής διεύθυνσης καθορίζουν τη διεύθυνση μέσα στη σελίδα και τα υπόλοιπα $36 - 12 = 24$ δυαδικά ψηφία της φυσικής διεύθυνσης αποτελούν τη φυσική διεύθυνση σελίδας, φδΣ.

Αφού η κρυφή μνήμη επεξεργαστή έχει χωρητικότητα 32 Kψηφιολέξεις με 16 ψηφιολέξεις ανά πλαίσιο και οργάνωση μονοσήμαντης απεικόνισης, θα έχει (32 Kψηφιολέξεις / 16 ψηφιολέξεις) πλαίσια = 2 K πλαίσια = $2 \times 2^{10} = 2^{11}$ πλαίσια. Επομένως τα 4 λιγότερο σημαντικά δυαδικά ψηφία της λογικής διεύθυνσης χρησιμοποιούνται για τον καθορισμό της διεύθυνσης της λέξης μέσα στο πλαίσιο, τα επόμενα 11 πιο σημαντικά δυαδικά ψηφία χρησιμοποιούνται για την διευθυνσιοδότηση του πλαισίου και τα υπόλοιπα $52 - (4 + 11) = 37$ δυαδικά ψηφία αποτελούν την ετικέτα.

Στο σχήμα 5.18.1 φαίνεται ο τρόπος προσπέλασης της κρυφής μνήμης πίνακα σελίδων και επεξεργαστή.



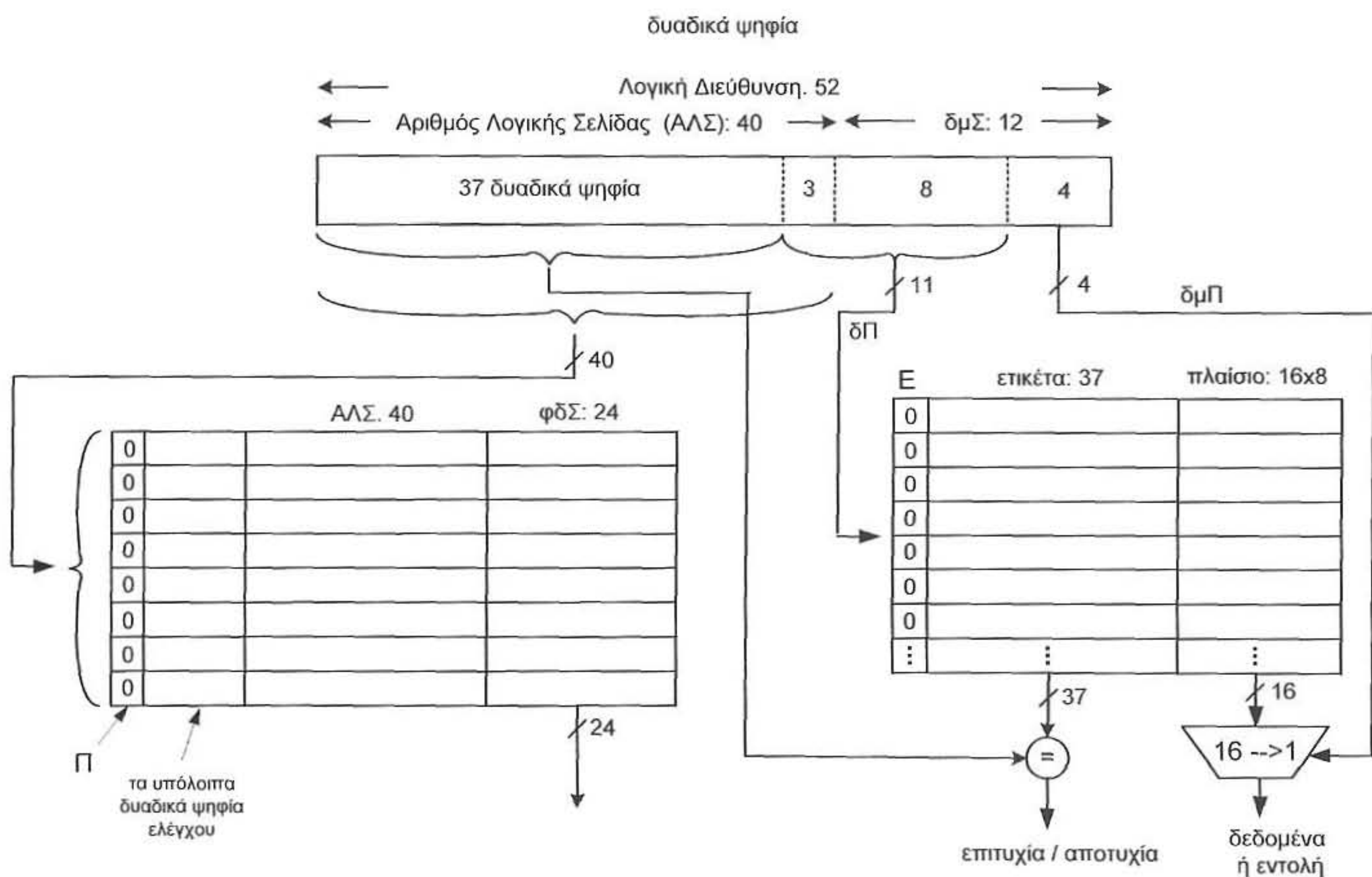
Σχήμα 5.18.1 Παράλληλη προσπέλαση κρυφής μνήμης πίνακα σελίδων και κρυφής μνήμης επεξεργαστή προσπελάσιμης με λογικές διευθύνσεις. δμΠ: διεύθυνση μέσα στο πλαίσιο, δΠ: διεύθυνση πλαισίου, Ε: δυαδικό ψηφίο εγκυρότητας.

Άσκηση 5.19

Να θεωρήσετε τον υπολογιστή της άσκησης 5. 18. Επίσης να θεωρήσετε ότι αρχικά η κρυφή μνήμη πίνακα σελίδων και η κρυφή μνήμη επεξεργαστή δεν περιέχουν χρήσιμη πληροφορία. Ποια θα είναι τα περιεχόμενά τους μετά κάθε μία από τις ακόλουθες διευθύνσεις $0100184C00BA1_{(16)}$, $0100AD4C00BF2_{(16)}$ και $0100184C56BA4_{(16)}$ που παράγει ο επεξεργαστής; Θεωρήστε ότι αυτές οι λογικές διευθύνσεις, μετά την προσκόμιση των απαιτούμενων σελίδων στην κύρια μνήμη, αντιστοιχούν στις φυσικές διευθύνσεις $0FF008BA1_{(16)}$, $000DA3BF2_{(16)}$ και $0A4529BA4_{(16)}$.

Απάντηση

Στο σχήμα 5.19.1 φαίνονται τα αρχικά περιεχόμενα της κρυφής μνήμης πίνακα σελίδων και της κρυφής μνήμης επεξεργαστή.



Σχήμα 5.19.1 Το δυαδικό ψηφίο παρουσίας Π κάθε θέσης της κρυφής μνήμης πίνακα σελίδων και το δυαδικό ψηφίο εγκυρότητας Ε κάθε θέσης της κρυφής μνήμης επεξεργαστή έχουν την τιμή 0. Επομένως αυτές οι δύο μνήμες δεν έχουν χρήσιμη πληροφορία.

α. Ο επεξεργαστής παράγει τη λογική διεύθυνση 0100184C00BA1₍₁₆₎.

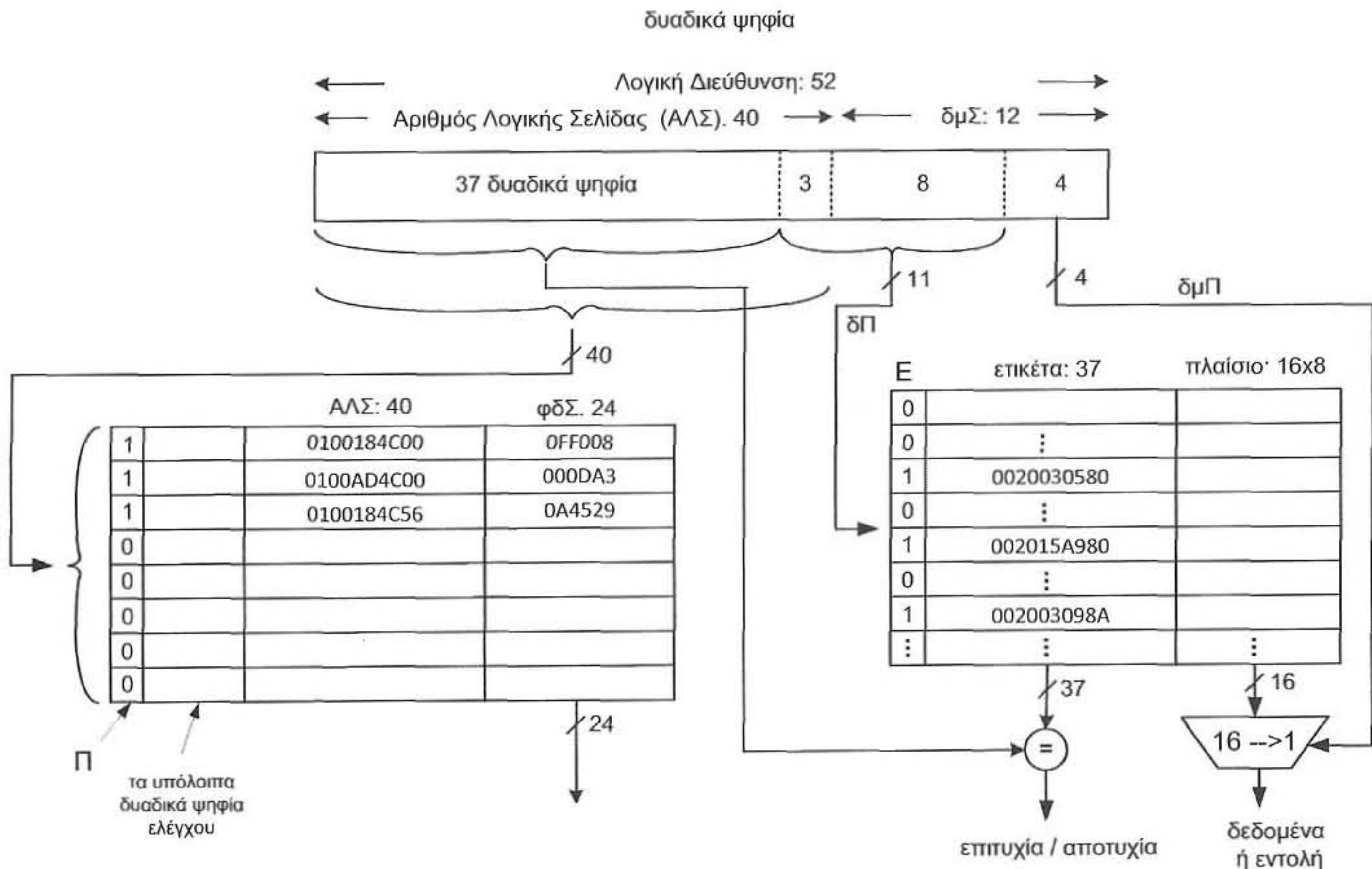
Τα 10 πιο σημαντικά δεκαεξαδικά ψηφία 0100184C00₍₁₆₎ (40 δυαδικά ψηφία) της λογικής διεύθυνσης που παράγει ο επεξεργαστής αποτελούν τον αριθμό λογικής σελίδας και χρησιμοποιούνται ως κλειδί για την προσπέλαση της κρυφής μνήμης πίνακα σελίδων, η οποία έχει οργάνωση πλήρους συσχέτισης. Επειδή το δυαδικό ψηφίο παρουσίας όλων των θέσεων της κρυφής μνήμης του πίνακα σελίδων έχει την τιμή 0, η ζητούμενη αντιστοίχιση αριθμός λογικής σελίδας, ΑΛΣ, φυσική διεύθυνση σελίδας, φδΣ, δεν βρίσκεται στην κρυφή μνήμη πίνακα σελίδων. Επομένως θα προσπελαστεί ο πίνακας σελίδων. Εάν στον πίνακα σελίδων βρεθεί η ζητούμενη αντιστοίχιση, τότε θα ενημερωθεί η κρυφή μνήμη πίνακα σελίδων. Εάν η ζητούμενη αντιστοίχιση δεν βρεθεί ούτε στον πίνακα σελίδων, τότε θα αναλάβει το λειτουργικό σύστημα να μεταφέρει τη ζητούμενη σελίδα από το δίσκο στην κύρια μνήμη και να ενημερώσει τόσο τον πίνακα σελίδων όσο και την κρυφή μνήμη πίνακα σελίδων.

Τα 11 δυαδικά ψηφία 00010111010 της λογικής διεύθυνσης, τα οποία δηλώνουν τη διεύθυνση του πλαισίου στην κρυφή μνήμη επεξεργαστή, θα χρησιμοποιηθούν για την προσπέλαση της κρυφής μνήμης επεξεργαστή. Επειδή όμως όλα τα δυαδικά ψηφία εγκυρότητας έχουν την τιμή 0 θα έχουμε αποτυχία. Εάν η σελίδα που περιέχει το ζητούμενο μπλοκ βρίσκεται στην κύρια μνήμη, ή εάν δε βρίσκεται στην κύρια μνήμη μετά τη μεταφορά της από το δίσκο στην κύρια μνήμη, ο ελεγκτής της κρυφής μνήμης επεξεργαστή θα μεταφέρει το ζητούμενο μπλοκ πληροφορίας από την κύρια μνήμη στην κρυφή μνήμη επεξεργαστή στο πλαίσιο με διεύθυνση 00010111010 και θα αποθηκεύσει στο αντίστοιχο πεδίο ετικέτας τα 37 πιο σημαντικά δυαδικά ψηφία της λογικής διεύθυνσης: 0 0000 0010 0000 0000 0011 0000 1001 1000 0000 (στο δεκαεξαδικό 0020030980₍₁₆₎)

Στο σχήμα 5.19.2 φαίνονται τα περιεχόμενα της κρυφής μνήμης πίνακα σελίδων και της κρυφής μνήμης επεξεργαστή μετά την προσπέλαση της πρώτης διεύθυνσης.

β. Ο επεξεργαστής παράγει τη λογική διεύθυνση 0100AD4C00BF2₍₁₆₎.

Τα 10 πιο σημαντικά δεκαεξαδικά ψηφία 0100AD4C00₍₁₆₎ (40 δυαδικά ψηφία) της λογικής διεύθυνσης που παράγει ο επεξεργαστής αποτελούν τον αριθμό λογικής σελίδας και χρησιμοποιούνται ως κλειδί για την προσπέλαση της κρυφής μνήμης πίνακα σελίδων, η οποία έχει οργάνωση πλήρους συσχέτισης. Επειδή δεν υπάρχει θέση της κρυφής μνήμης του πίνακα σελίδων που το δυαδικό ψηφίο παρουσίας να έχει την τιμή 1 και το πεδίο ΑΛΣ να έχει την τιμή 0100AD4C00₍₁₆₎, η ζητούμενη αντιστοίχιση (ΑΛΣ, φδΣ) δεν βρίσκεται στην κρυφή μνήμη πίνακα σελίδων. Επομένως θα προσπελαστεί ο πίνακας σελίδων. Εάν στον πίνακα σελίδων βρεθεί η ζητούμενη αντιστοίχιση, τότε θα ενημερωθεί η κρυφή μνήμη πίνακα σελίδων. Εάν η ζητούμενη αντιστοίχιση δεν βρεθεί ούτε στον



Σχήμα 5.19.4

Άσκηση 5.20 (5.13 βιβλίου)

Θεωρούμε υπολογιστή με ιδεατή μνήμη (virtual memory) που υλοποιείται με την τεχνική της σελιδοποίησης. Θεωρούμε ότι υπάρχει κρυφή μνήμη πίνακα σελίδων (TLB) των 8 θέσεων με οργάνωση πλήρους συσχέτισης και υβριδική κρυφή μνήμη επεξεργαστή (CPU cache memory). Η λογική διεύθυνση είναι των 52 δυαδικών ψηφίων, ενώ η φυσική διεύθυνση των 36 δυαδικών ψηφίων και το μέγεθος της σελίδας ίσο με 4 Kψηφιολέξεις (Kbytes). Η κύρια μνήμη έχει οργάνωση μίας ψηφιολέξης ανά θέση. Να σχεδιάσετε το κατάλληλο σχήμα που να δείχνει πως χρησιμοποιείται η λογική διεύθυνση για την προσπέλαση της κρυφής μνήμης πίνακα σελίδων και επεξεργαστή για κάθε μία των κάτωθι περιπτώσεων. Η κρυφή μνήμη επεξεργαστή σε κάθε περίπτωση να έχει το ελάχιστο κόστος υλοποίησης.

α. Η κρυφή μνήμη επεξεργαστή έχει χωρητικότητα 4 Kψηφιολέξεις με 16 ψηφιολέξεις ανά πλαίσιο.

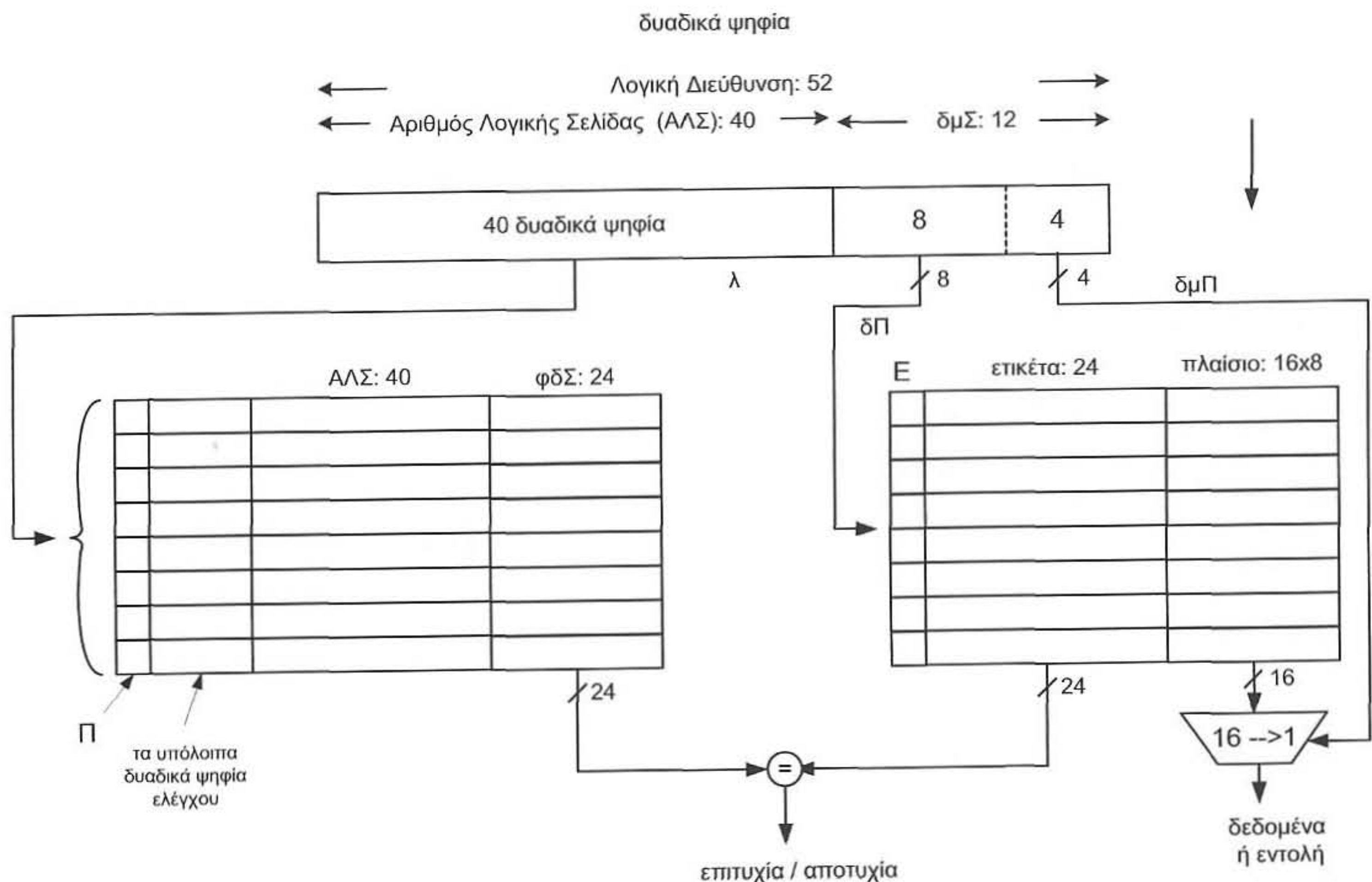
β. Η κρυφή μνήμη επεξεργαστή έχει χωρητικότητα 8 Kψηφιολέξεις με 16 ψηφιολέξεις ανά πλαίσιο.

Απάντηση

Η κρυφή μνήμη με οργάνωση μονοσήμαντης απεικόνισης καθώς και η κρυφή μνήμη με οργάνωση πλήρους συσχέτισης είναι ειδικές περιπτώσεις της κρυφής μνήμης με οργάνωση τ-τρόπων συνόλου συσχέτισης. Για $\tau=1$ παίρνουμε την κρυφή μνήμη με οργάνωση μονοσήμαντης απεικόνισης, ενώ για τ ίσο με το πλήθος των πλαισίων της κρυφής μνήμης παίρνουμε την κρυφή μνήμη με οργάνωση πλήρους συσχέτισης. Το κόστος υλοποίησης μίας κρυφής μνήμης τ-τρόπων συνόλου συσχέτισης αυξάνεται καθώς αυξάνεται η τιμή του τ , εφόσον τα υπόλοιπα χαρακτηριστικά και η τεχνολογία υλοποίησης παραμένουν τα ίδια. Επομένως σύμφωνα με την άσκηση θα πρέπει να χρησιμοποιήσουμε σε κάθε περίπτωση την ελάχιστη τιμή του τ .

Λαμβάνοντας υπόψη ότι το μέγεθος της σελίδας είναι 4 Kψηφιολέξεις $= 2^{12}$ ψηφιολέξεις, συμπεραίνουμε ότι τα 12 λιγότερο σημαντικά δυαδικά ψηφία της λογικής διεύθυνσης καθορίζουν τη διεύθυνση μέσα στη σελίδα και επομένως δεν επηρεάζονται από τη μετάφραση σε φυσική διεύθυνση.

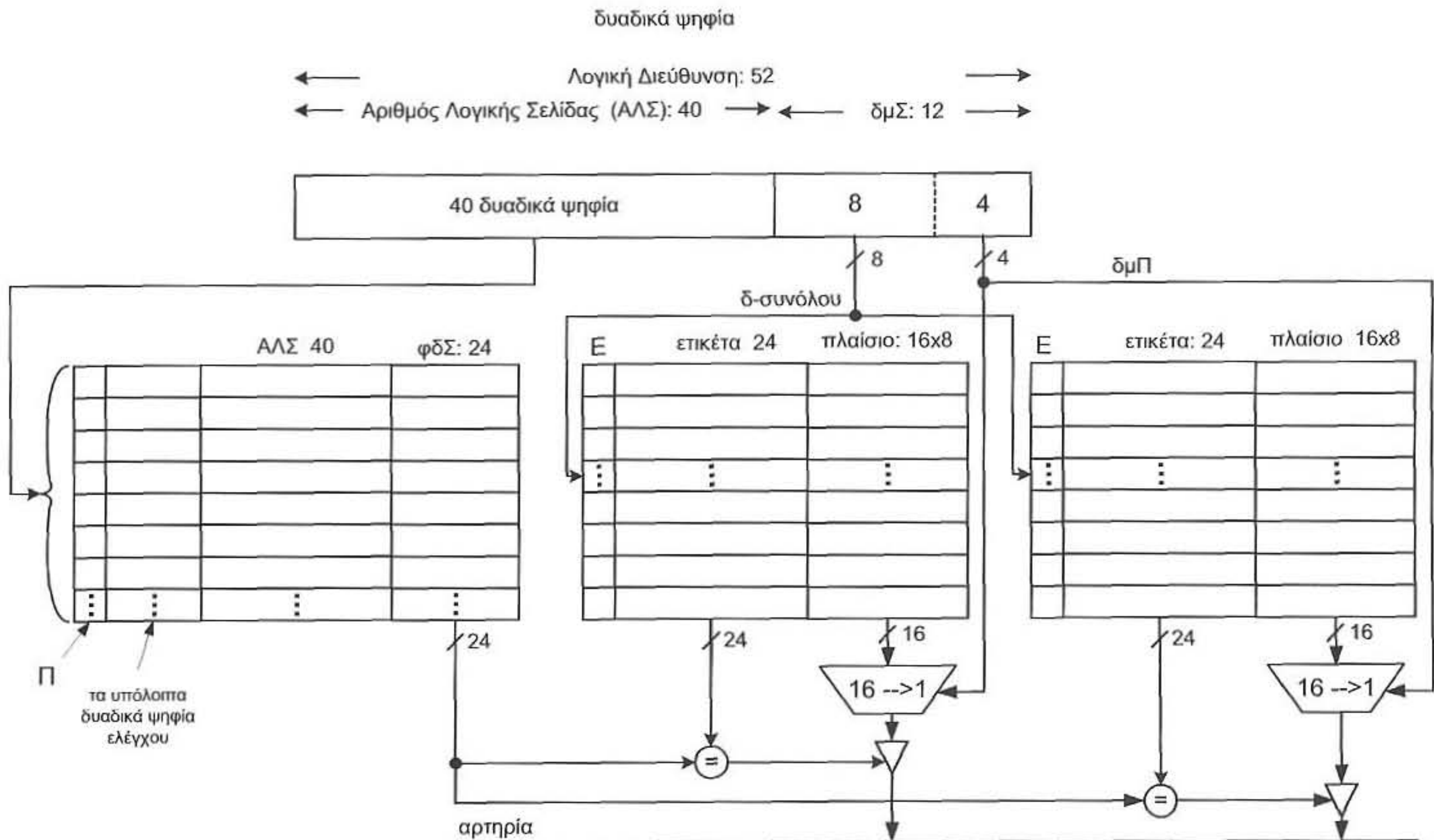
α. Η κρυφή μνήμη επεξεργαστή έχει 16 ψηφιολέξεις ανά πλαίσιο, άρα τα 4 λιγότερο σημαντικά δυαδικά ψηφία θα δείχνουν τη διεύθυνση της λέξης μέσα στο πλαίσιο. Αφού η κρυφή μνήμη επεξεργαστή είναι των 4 Kψηφιολέξεων (Kbytes) με 16 ψηφιολέξεις ανά πλαίσιο η κρυφή μνήμη επεξεργαστή θα έχει $4 \text{ Kψηφιολέξεις} / 16 \text{ ψηφιολέξεις} = 4 \times 2^{10} \text{ ψηφιολέξεις} / 2^4 \text{ ψηφιολέξεις} = 2^8 = 256$ πλαίσια. Εάν η κρυφή μνήμη επεξεργαστή έχει οργάνωση μονοσήμαντης απεικόνισης τότε τα επόμενα 8 πιο σημαντικά δυαδικά ψηφία θα αποτελούν την διεύθυνση του πλαισίου, επομένως για την προσπέλαση της κρυφής μνήμης επεξεργαστή απαιτούνται τα 12 λιγότερο σημαντικά δυαδικά ψηφία της λογικής διεύθυνσης, τα οποία όπως είδαμε δε επηρεάζονται από τη μετάφραση της λογικής διεύθυνσης σε φυσική. Αφού όλα τα δυαδικά ψηφία του πεδίου δμΣ χρησιμοποιούνται από τα πεδία διεύθυνση μέσα στο πλαίσιο δμΠ και διεύθυνση πλαισίου δΠ, με κάθε πλαίσιο της κρυφής μνήμης επεξεργαστή αποθηκεύεται μόνο μια ετικέτα, η φυσική διεύθυνση της σελίδας. Επειδή η φυσική διεύθυνση είναι των 36 δυαδικών ψηφίων και το μέγεθος της σελίδας 4 Kψηφιολέξεις $= 2^{12}$ ψηφιολέξεις, συνεπάγεται ότι η φυσική διεύθυνση σελίδας είναι των 24 δυαδικών ψηφίων. Το σχήμα 5.20.1 δείχνει τον τρόπο χρησιμοποίησης της λογικής διεύθυνσης για την προσπέλαση της κρυφής μνήμης πίνακα σελίδων και επεξεργαστή.



Σχήμα 5.20.1

β. Η κρυφή μνήμη επεξεργαστή έχει 16 ψηφιολέξεις ανά πλαίσιο, άρα τα 4 λιγότερο σημαντικά δυαδικά ψηφία θα δείχνουν τη διεύθυνση της λέξης μέσα στο πλαίσιο. Αφού η κρυφή μνήμη επεξεργαστή είναι των 8 Kψηφιολέξεων (Kbytes) με 16 ψηφιολέξεις ανά πλαίσιο η κρυφή μνήμη επεξεργαστή θα έχει $8 \text{ Kψηφιολέξεις} / 16 \text{ ψηφιολέξεις} = 2^3 \times 2^{10}$ ψηφιολέξεις / 2^4 ψηφιολέξεις = 2^9 πλαίσια = 512 πλαίσια. Εάν η κρυφή μνήμη επεξεργαστή έχει άμεση οργάνωση, τότε τα επόμενα 9 πιο σημαντικά δυαδικά ψηφία θα αποτελούν τη διεύθυνση του πλαισίου, επομένως για την προσπέλαση της κρυφής μνήμης επεξεργαστή απαιτούνται τα 13 λιγότερο σημαντικά δυαδικά ψηφία της λογικής διεύθυνσης. Όπως είδαμε μόνο τα 12 λιγότερο σημαντικά δυαδικά ψηφία της λογικής διεύθυνσης δεν επηρεάζονται από τη μετάφραση της λογικής διεύθυνσης σε φυσική. Επομένως η κρυφή μνήμη επεξεργαστή δε μπορεί να έχει άμεση οργάνωση. Εάν η κρυφή μνήμη επεξεργαστή έχει οργάνωση δύο τρόπων συνόλου συσχέτισης τότε θα περιέχει $2^9 / 2 = 2^8 = 256$ σύνολα, άρα θα απαιτούνται 8 δυαδικά ψηφία για τον καθορισμό του συνόλου και συνολικά τα 12 λιγότερο σημαντικά δυαδικά ψηφία της λογικής διεύθυνσης για την προσπέλαση της κρυφής μνήμης επεξεργαστή, τα οποία όπως είδαμε δε επηρεάζονται από την μετάφραση της λογικής διεύθυνσης σε φυσική.

Το σχήμα 5.20.2 δείχνει τον τρόπο χρησιμοποίησης της λογικής διεύθυνσης για την προσπέλαση της κρυφής μνήμης πίνακα σελίδων και επεξεργαστή.



Σχήμα 5.20.2

Η φυσική διεύθυνση της σελίδας είναι και πάλι των 24 δυαδικών ψηφίων. Για τον ίδιο λόγο, όπως και στην προηγούμενη περίπτωση, με κάθε πλαίσιο της κρυφής μνήμης επεξεργαστή αποθηκεύεται μόνο μια ετικέτα, η φυσική διεύθυνση της σελίδας.

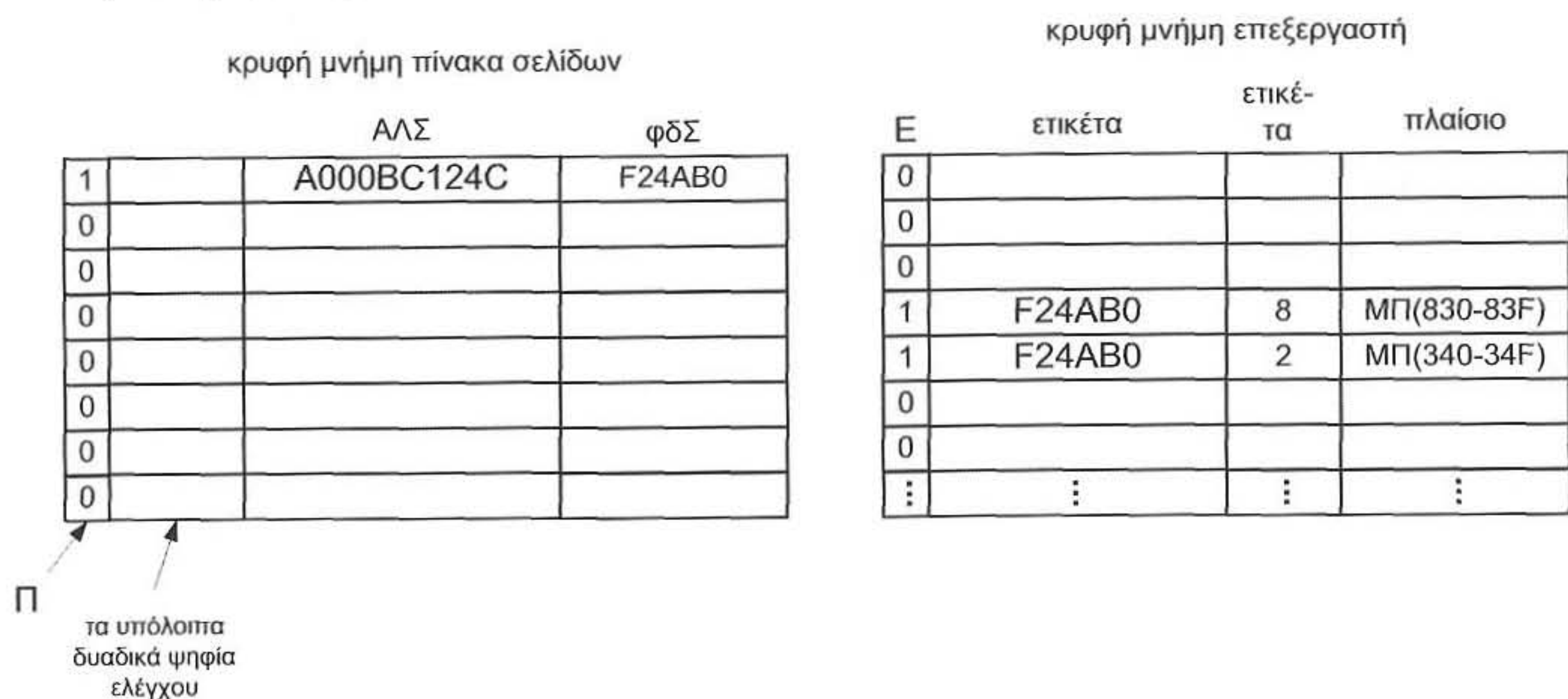
Άσκηση 5.21 (5.14 βιβλίου)

Θεωρούμε υπολογιστή με ιδεατή μνήμη (virtual memory) που υλοποιείται με την τεχνική της σελιδοποίησης. Θεωρούμε ότι υπάρχει κρυφή μνήμη πίνακα σελίδων (TLB) των 8 θέσεων με οργάνωση πλήρους συσχέτισης και υβριδική κρυφή μνήμη επεξεργαστή (CPU cache memory) με οργάνωση μονοσήμαντης απεικόνισης. Η λογική διεύθυνση είναι των 52 δυαδικών ψηφίων ενώ η φυσική διεύθυνση των 36 δυαδικών ψηφίων και το μέγεθος της σελίδας ίσο με 4 Kψηφιολέξεις (Kbytes). Η κρυφή μνήμη επεξεργαστή αποτελείται από 16 πλαίσια των 16 ψηφιολέξεων το κάθε ένα. Η κύρια μνήμη έχει οργάνωση μίας ψηφιολέξης ανά θέση. Τα τρέχοντα περιεχόμενα της κρυφής μνήμης πίνακα σελίδων και επεξεργαστή δίνονται στο σχήμα 5.21.1.

α. Να σχεδιάσετε το κατάλληλο σχήμα το οποίο να δείχνει πως χρησιμοποιείται η λογική διεύθυνση για την προσπέλαση της κρυφής μνήμης πίνακα σελίδων και επεξεργαστή.

β. Υποθέστε ότι ο επεξεργαστής παράγει τις ακόλουθες λογικές διευθύνσεις στο δεκαεξαδικό: A000BC124C83A, A000BC124C341, B000000011201.

Να περιγράψετε τι θα συμβεί σε κάθε περίπτωση και στην περίπτωση αποτυχίας (miss) να δώσετε τα περιεχόμενα της κρυφής μνήμης πίνακα σελίδων και επεξεργαστή μετά την ενημέρωση.

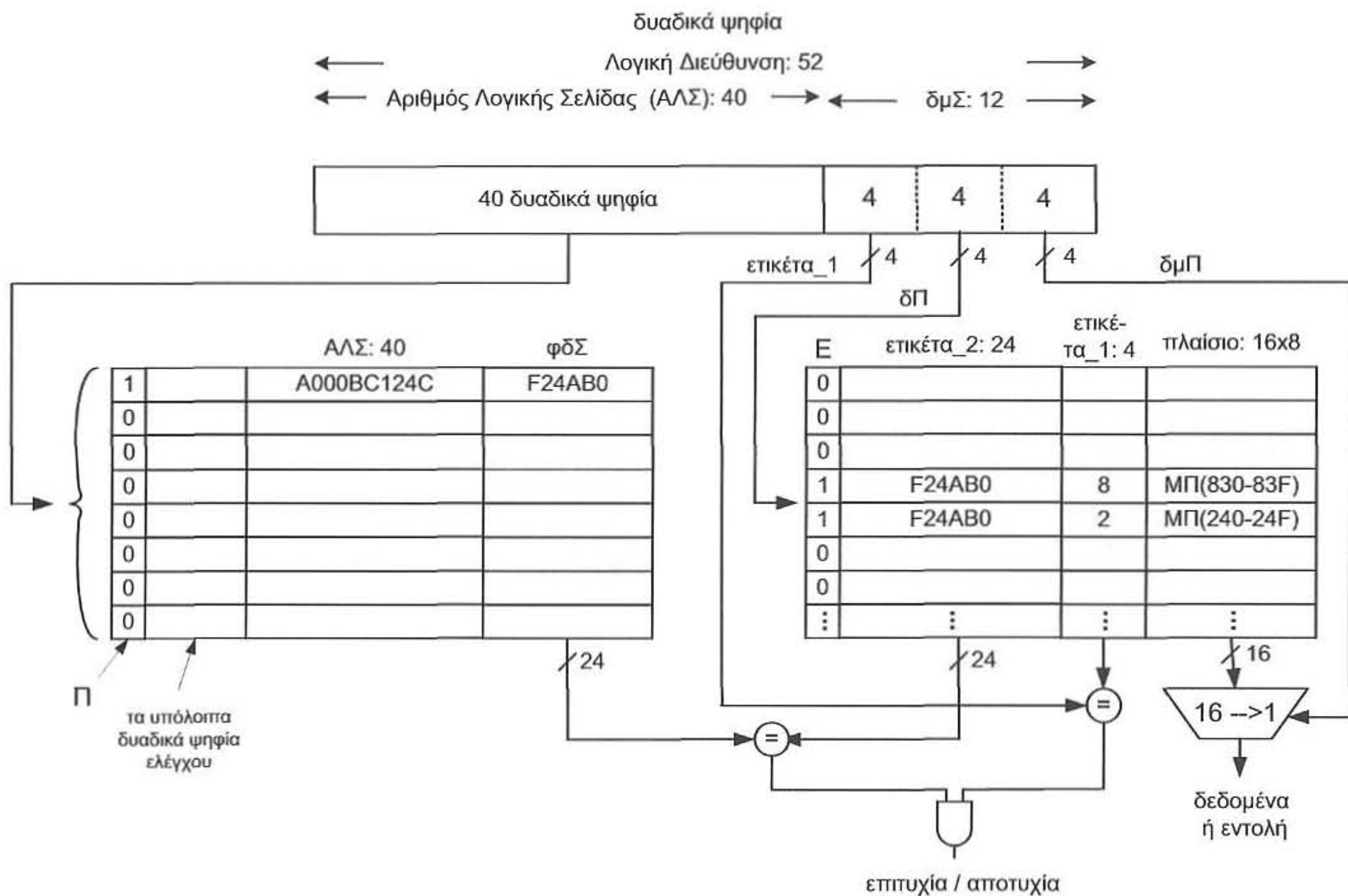


Σχήμα 5.21.1

Απάντηση

α. Λαμβάνοντας υπόψη ότι η λογική διεύθυνση είναι των 52 δυαδικών ψηφίων, η κύρια μνήμη έχει οργάνωση μίας ψηφιολέξης ανά θέση μνήμης και το μέγεθος της σελίδας είναι 4 Kψηφιολέξεις, συμπεραίνουμε ότι ο χώρος λογικών διευθύνσεων αποτελείται από $(2^{52} \text{ ψηφιολέξεις}) / (4 \text{ Kψηφιολέξεις}) = 2^{52} / 2^{12} = 2^{40}$ σελίδες. Άρα ο Αριθμός Λογικής Σελίδας, ΑΛΣ, είναι των 40 δυαδικών ψηφίων και η διεύθυνση μέσα στη σελίδα, δμΣ, είναι των 12 δυαδικών ψηφίων. Ο χώρος φυσικών διευθύνσεων αποτελείται από $(2^{36} \text{ ψηφιολέξεις}) / (4 \text{ Kψηφιολέξεις}) = 2^{36} / 2^{12} = 2^{24}$ σελίδες. Άρα η φυσική διεύθυνση σελίδας είναι των 24 δυαδικών ψηφίων. Η κρυφή μνήμη επεξεργαστή είναι υβριδική, δηλαδή για την προσπέλασή της χρησιμοποιείται μόνον το τμήμα της λογικής διεύθυνσης που αποτελεί τη διεύθυνση μέσα στη σελίδα, δμΣ. Οπότε λαμβάνοντας υπόψη ότι η κρυφή μνήμη επεξεργαστή αποτελείται από 16 πλαίσια των 16 ψηφιολέξεων το κάθε ένα, συμπεραίνουμε ότι από τα 12 δυαδικά ψηφία της δμΣ τα 4 λιγότερο σημαντικά χρησιμοποιούνται για τον καθορισμό της λέξης μέσα στο πλαίσιο, τα επόμενα 4 πιο σημαντικά δυαδικά ψηφία χρησιμοποιούνται για την επιλογή του πλαισίου και τα υπόλοιπα 4 πιο σημαντικά

δυναδικά ψηφία αποτελούν την ετικέτα, η οποία είναι διαφορετική για τα μπλοκ της σελίδας που αντιστοιχούν στο ίδιο πλαίσιο. Στο σχήμα 5.21.2 την ετικέτα αυτή την ονομάζουμε ετικέτα_1. Λόγω του γεγονότος ότι η κρυφή μνήμη επεξεργαστή είναι υβριδική, με κάθε πλαίσιο είναι αποθηκευμένη και μία ετικέτα, η ετικέτα_2 στο σχήμα 5.21.2, η οποία αποτελεί τη φυσική διεύθυνση της σελίδας στην οποία ανήκει το συγκεκριμένο πλαίσιο. Στο σχήμα 5.21.2 δίνεται ο τρόπος που χρησιμοποιούνται τα δυναδικά ψηφία της λογικής διεύθυνσης για την προσπέλαση της κρυφής μνήμης του πίνακα σελίδων και της υβριδικής κρυφής μνήμης του επεξεργαστή.

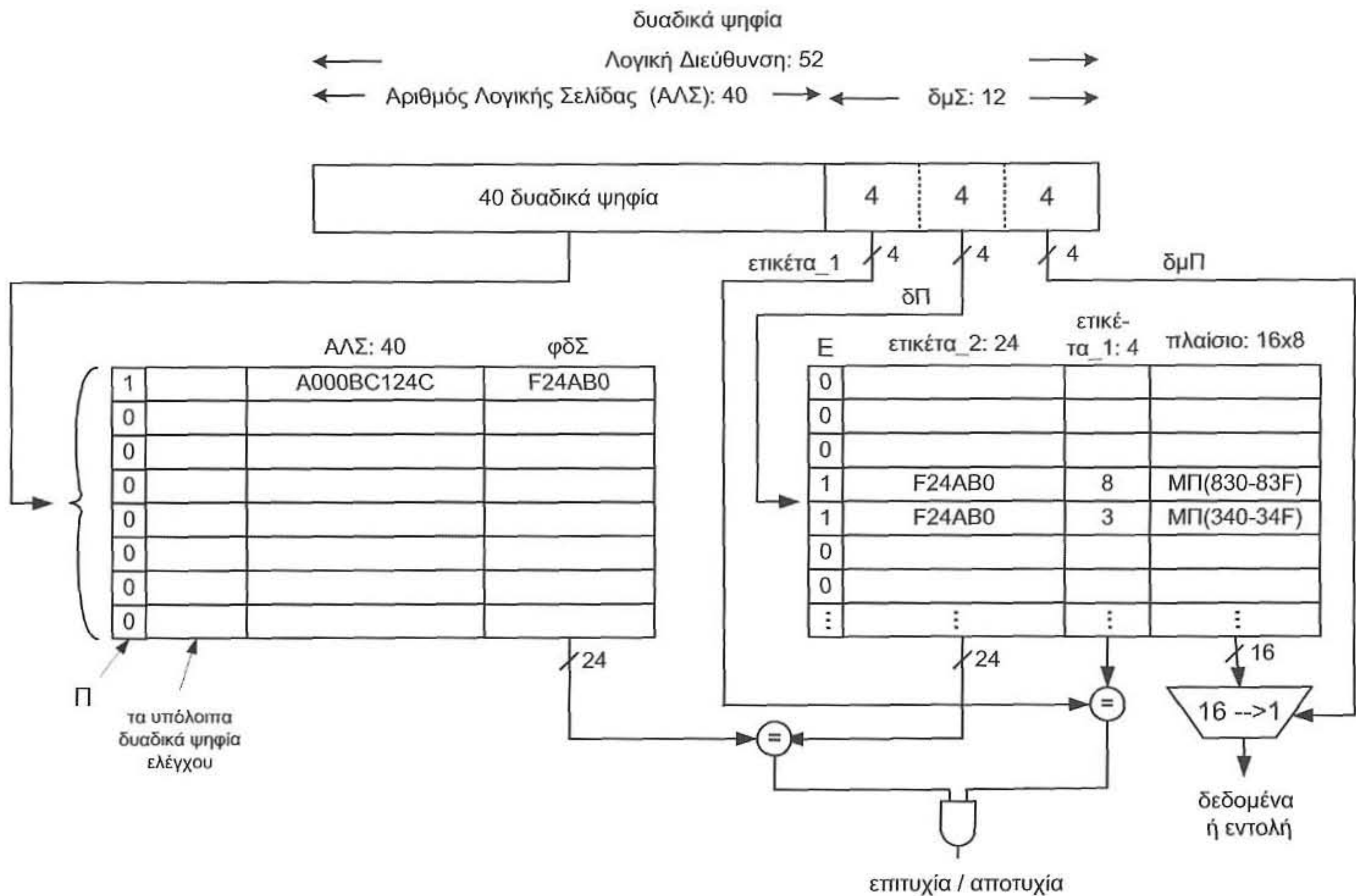


Σχήμα 5.21.2 ΜΠ(X-Y): δηλώνει ότι το συγκεκριμένο πλαίσιο περιέχει τα δεδομένα ή εντολές των διευθύνσεων X έως και Y της σελίδας που δηλώνεται στο πεδίο ετικέτα_2. Ως παράδειγμα αναφέρουμε ότι στο σχήμα το ΜΠ(830-83F) δηλώνει ότι το συγκεκριμένο πλαίσιο περιέχει τα δεδομένα ή εντολές των διευθύνσεων F24AB0830- F24AB083F.

β.1. Όταν ο επεξεργαστής παράγει τη λογική διεύθυνση A000BC124C83A, τα 40 πιο σημαντικά δυναδικά ψηφία της, A000BC124C, που αποτελούν τον ΑΛΣ χρησιμοποιούνται ως κλειδί για να προσπελαστεί η κρυφή μνήμη πίνακα σελίδων, η οποία στη συγκεκριμένη περίπτωση είναι πλήρους συσχέτισης. Παρατηρούμε ότι στην πρώτη θέση της κρυφής μνήμης πίνακα σελίδων στο πεδίο ΑΛΣ υπάρχει η ζητούμενη τιμή. Επίσης το

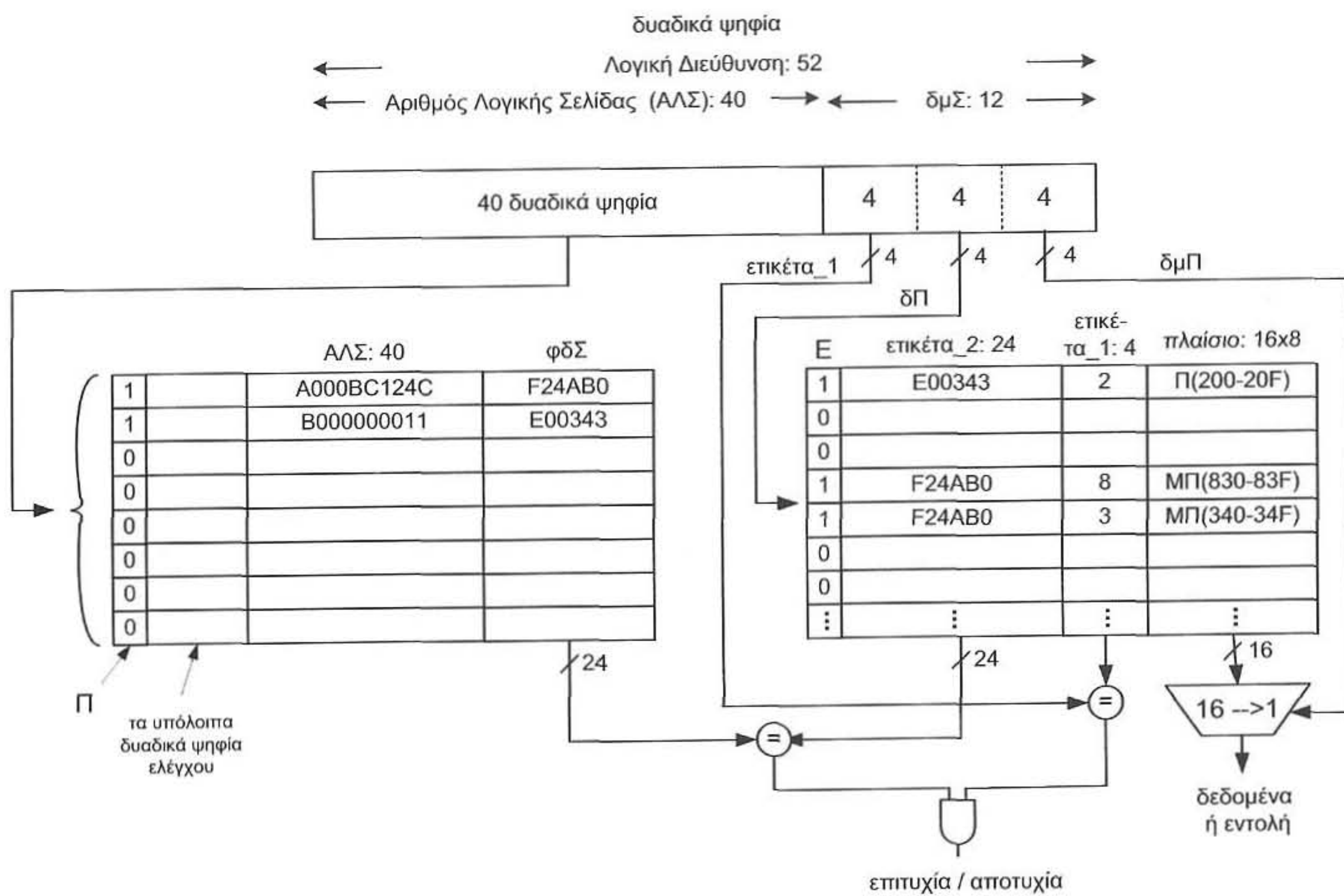
δυαδικό ψηφίο Π έχει τη λογική τιμή 1, που σημαίνει ότι η πληροφορία η οποία υπάρχει σ' αυτή τη θέση της κρυφής μνήμης ισχύει. Επομένως το περιεχόμενο του πεδίου φδΣ της ίδιας θέσης προσπελαύνεται. Ταυτόχρονα τα υπόλοιπα 12 λιγότερο σημαντικά δυαδικά ψηφία, 83Α, της λογικής διεύθυνσης, τα οποία αποτελούν την δμΣ, χρησιμοποιούνται για την προσπέλαση της κρυφής μνήμης επεξεργαστή. Επειδή το πεδίο διεύθυνσης πλαισίου δΠ περιέχει τον αριθμό 3 προσπελαύνεται η θέση της κρυφής μνήμης επεξεργαστή με διεύθυνση 3. Το περιεχόμενο του πεδίου ετικέτα_1 των 4 δυαδικών ψηφίων της δμΣ (8) συγκρίνεται με το περιεχόμενο του πεδίου ετικέτα_1 (8) της θέσης με διεύθυνση 3 της κρυφής μνήμης επεξεργαστή. Παρατηρούμε ότι τα περιεχόμενα είναι ταυτοτικά (είναι το 8). Ταυτόχρονα η φδΣ που διαβάστηκε από την κρυφή μνήμη του πίνακα σελίδων συγκρίνεται με την ετικέτα_2 που διαβάστηκε από τη θέση της κρυφής μνήμης με διεύθυνση 3. Παρατηρούμε και πάλι ότι τα περιεχόμενα είναι ταυτοτικά (είναι το F24AB0), επομένως έχουμε επιτυχία και τα δεδομένα ή εντολή που εμφανίζονται στην έξοδο του πολύπλεκτη είναι τα ζητούμενα.

β.2. Όταν ο επεξεργαστής παράγει τη λογική διεύθυνση A000BC124C341 τα 40 πιο σημαντικά δυαδικά ψηφία της, A000BC124C, που αποτελούν τον ΑΛΣ χρησιμοποιούνται για να προσπελαστεί η κρυφή μνήμη πίνακα σελίδων. Παρατηρούμε ότι στην πρώτη θέση της κρυφής μνήμης πίνακα σελίδων στο πεδίο ΑΛΣ υπάρχει η ζητούμενη τιμή, επομένως το περιεχόμενο του πεδίου φδΣ της ίδιας θέσης προσπελαύνεται. Ταυτόχρονα τα υπόλοιπα 12 λιγότερο σημαντικά δυαδικά ψηφία, 341₍₁₆₎, της λογικής διεύθυνσης, τα οποία αποτελούν την δμΣ, χρησιμοποιούνται για την προσπέλαση της κρυφής μνήμης επεξεργαστή. Επειδή το πεδίο διεύθυνσης πλαισίου, δΠ, περιέχει τον αριθμό 4, προσπελαύνεται η θέση της κρυφής μνήμης επεξεργαστή με διεύθυνση 4. Το περιεχόμενο του πεδίου ετικέτα_1 των 4 δυαδικών ψηφίων της δμΣ συγκρίνεται με το περιεχόμενο του πεδίου ετικέτα_1 της θέσης με διεύθυνση 4 της κρυφής μνήμης επεξεργαστή. Παρατηρούμε ότι τα περιεχόμενα είναι διαφορετικά. Επομένως το πλαίσιο που είναι αποθηκευμένο στη θέση 4 της κρυφής μνήμης επεξεργαστή δεν είναι το ζητούμενο. Ταυτόχρονα η φδΣ που διαβάστηκε από την κρυφή μνήμη του πίνακα σελίδων συγκρίνεται με την ετικέτα_2 που διαβάστηκε από τη θέση της κρυφής μνήμης με διεύθυνση 4 και παρατηρούμε ότι υπάρχει ταυτότητα. Επομένως η θέση της κρυφής μνήμης επεξεργαστή με διεύθυνση 4 περιέχει ένα μπλοκ της σελίδας που περιέχει την ζητούμενη πληροφορία, αλλά όχι το μπλοκ που περιέχει την ζητούμενη πληροφορία. Τότε μεταφέρεται το περιεχόμενο των φυσικών διευθύνσεων F24AB0340- F24AB034F από την κύρια μνήμη στο πλαίσιο της κρυφής μνήμης επεξεργαστή με διεύθυνση 4 και ενημερώνεται κατάλληλα η ετικέτα_1 όπως φαίνεται στο σχήμα 5.21.3.



Σχήμα 5.21.3

β.3. Όταν ο επεξεργαστής παράγει τη λογική διεύθυνση B000000011201, τα 40 πιο σημαντικά δυαδικά ψηφία της B000000011₍₁₆₎, τα οποία αποτελούν τον ΑΛΣ, χρησιμοποιούνται για να προσπελαστεί η κρυφή μνήμη πίνακα σελίδων. Παρατηρούμε ότι εκτός της πρώτη θέσης της κρυφής μνήμης πίνακα σελίδων, στο πεδίο ΑΛΣ της οποίας υπάρχει το A000BC124C, σε όλες τις άλλες θέσεις το πεδίο παρουσίας έχει την τιμή 0, επομένως αυτές οι θέσεις δεν περιέχουν χρήσιμη πληροφορία αντιστοίχισης (ΑΛΣ, δφΣ). Θα πρέπει επομένως να προσπελαστεί ο Πίνακας Σελίδων. Εάν ο Πίνακας Σελίδων περιέχει την ζητούμενη αντιστοίχιση, τότε η ζητούμενη πληροφορία υπάρχει στην κύρια μνήμη, τότε ενημερώνεται κατάλληλα η κρυφή μνήμη πίνακα σελίδων. Επίσης το περιεχόμενο του ζητούμενου μπλοκ μεταφέρεται στο αντίστοιχο πλαίσιο της κρυφής μνήμης επεξεργαστή. Εάν η ζητούμενη σελίδα δεν υπάρχει στην κύρια μνήμη, τότε το λειτουργικό σύστημα θα την φέρει από το δίσκο στην κύρια μνήμη και μετά θα ενημερώσει τον πίνακα σελίδων, την κρυφή μνήμη πίνακα σελίδων και την κρυφή μνήμη επεξεργαστή. Τα περιεχόμενα της κρυφής μνήμης πίνακα σελίδων και επεξεργαστή μετά την ενημέρωση φαίνονται στο σχήμα 5.21.4. Υποθέσαμε ότι το λειτουργικό σύστημα τοποθέτησε τη σελίδα στο πλαίσιο με φυσική διεύθυνση E00343.



Σχήμα 5.21.4

ΚΕΦΑΛΑΙΟ

6

Ασκήσεις για το σύστημα διασύνδεσης και τη διαδικασία εισόδου - εξόδου

Άσκηση 6.1 (6.1 βιβλίου)

Θεωρήστε μια αρτηρία που χρησιμοποιείται για τη μεταφορά πληροφορίας 4 ψηφιο-λέξεων κάθε φορά και ότι, ανάλογα με τις μονάδες που επικοινωνούν, η μεταφορά από τον υπηρέτη (slave) στον κύριο (master) διαρκεί στο 20% των περιπτώσεων 10 nsec, στο 30% των περιπτώσεων 20 nsec, στο 40% των περιπτώσεων 30 nsec και στο 10% των περιπτώσεων 60 nsec. Θεωρήστε επίσης ότι η διαιτησία βάζει καθυστέρηση 5 nsec. Υπολογίστε το μέσο ρυθμό μεταφοράς πληροφορίας σε ψηφιολέξεις ανά δευτερόλεπτο για κάθε μια από τις ακόλουθες περιπτώσεις.

- α. Η αρτηρία είναι σύγχρονη με περίοδο του σήματος χρονισμού (ρολογιού) 5 nsec
 - i. χωρίς δυνατότητα εισαγωγής κύκλων ρολογιού καθυστέρησης (wait states)
 - ii. με δυνατότητα εισαγωγής κύκλων ρολογιού καθυστέρησης
- β. Η αρτηρία είναι ασύγχρονη και απαιτεί 10 nsec επιπλέον για κάθε μεταφορά λόγω του πρωτοκόλλου χειραψίας.

Απάντηση

α. i. Για μεταφορά πληροφορίας από ή προς την αργότερη μονάδα εισ/εξ απαιτούνται 5 nsec για την διαιτησία και 60 nsec για τη μεταφορά, δηλαδή συνολικά 65 nsec ή 13 κύκλοι του σήματος χρονισμού. Άρα ο ρυθμός μεταφοράς δεδομένων θα είναι:
$$(4 \text{ ψηφιολέξεις}) / (65 \text{ nsec}) = (4 \times 10^9) / 65 \text{ ψηφιολέξεις/δευτερόλεπτο} = 61,5 \times 10^6 \text{ ψηφιολέξεις/δευτερόλεπτο} = 61,5 \text{ Mψηφιολέξεις/δευτερόλεπτο}.$$

α. ii. Λαμβάνοντας υπόψη τη διαιτησία συμπεραίνουμε ότι στο 20% των περιπτώσεων η μεταφορά κοστίζει 3 κύκλους του σήματος χρονισμού, στο 30% περιπτώσεων κοστίζει 5 κύκλους, στο 40% περιπτώσεων κοστίζει 7 κύκλους, και στο 10% των περιπτώσεων κοστίζει 13 κύκλους. Επομένως ο μέσος αριθμός κύκλων ρολογιού που απαιτούνται για μια μεταφορά είναι $0,2 \times 3 + 0,3 \times 5 + 0,4 \times 7 + 0,1 \times 13 = 6,2$ κύκλοι και ο μέσος ρυθμός μεταφοράς:

$$(4 \text{ ψηφιολέξεις}) / (6,2 \times 5 \text{ nsec}) = 4 / (6,2 \times 5 \times 10^{-9}) \text{ ψηφιολέξεις/δευτερόλεπτο} = 129 \text{ Mψηφιολέξεις/δευτερόλεπτο}.$$

β. Τότε ο μέσος χρόνος που απαιτείται για μια μεταφορά είναι:

$$0,2 \times (10+10+5) + 0,3 \times (20+10+5) + 0,4 \times (30+10+5) + 0,1 \times (60+10+5) = 41 \text{ nsec.}$$

Επομένως ο ρυθμός μεταφοράς είναι: $4 \text{ ψηφιολέξεις} / 41 \text{ nsec} = 97,56 \text{ Mψηφιολέξεις/δευτερόλεπτο.}$

Παρατηρούμε ότι η σύγχρονη αρτηρία με δυνατότητα εισαγωγής κύκλων καθυστέρησης επιτυγχάνει το μεγαλύτερο μέσο ρυθμό μεταφοράς πληροφορίας.

Άσκηση 6.2

Θεωρήστε μια αρτηρία που χρησιμοποιείται για τη μεταφορά πληροφορίας 8 ψηφιολέξεων κάθε φορά και ότι, ανάλογα με τις μονάδες που επικοινωνούν, η μεταφορά από τον υπηρέτη (slave) στον κύριο (master) διαρκεί στο 20% των περιπτώσεων 12 nsec, στο 30% των περιπτώσεων 19 nsec, στο 40% των περιπτώσεων 28 nsec και στο 10% των περιπτώσεων 62 nsec. Θεωρήστε επίσης ότι η διαιτησία βάζει καθυστέρηση 5 nsec. Υπολογίστε το μέσο ρυθμό μεταφοράς πληροφορίας ανά δευτερόλεπτο για κάθε μια από τις ακόλουθες περιπτώσεις.

α. Η αρτηρία είναι σύγχρονη με περίοδο του σήματος χρονισμού (ρολογιού) 5 nsec

- i. χωρίς δυνατότητα εισαγωγής κύκλων ρολογιού καθυστέρησης (wait states)
- ii. με δυνατότητα εισαγωγής κύκλων ρολογιού καθυστέρησης

β. Η αρτηρία είναι ασύγχρονη και για κάθε μεταφορά, λόγω του πρωτοκόλλου χειραψίας, απαιτεί 10 nsec επιπλέον.

Απάντηση

α. i. Αφού η διαιτησία βάζει καθυστέρηση 5 nsec και η περίοδος του σήματος χρονισμού είναι 5 nsec, για τη διαιτησία απαιτείται ένας κύκλος ρολογιού. Για τη μεταφορά πληροφορίας από ή προς την αργότερη μονάδα εισ/εξ απαιτούνται 62 nsec, αλλά επειδή στη σύγχρονη αρτηρία για τη μεταφορά πληροφορίας διατίθεται ένα ακέραιο πλήθος κύκλων ρολογιού, απαιτούνται 13 κύκλοι του σήματος χρονισμού. Άρα ο ρυθμός μεταφοράς δεδομένων θα είναι:

$$(8 \text{ ψηφιολέξεις}) / ((1+13) \times 5 \text{ nsec}) = (8 \times 10^9) / 70 \text{ ψηφιολέξεις/δευτερόλεπτο} = 114,28 \times 10^6 \text{ ψηφιολέξεις/δευτερόλεπτο} = 114,28 \text{ Mψηφιολέξεις/δευτερόλεπτο.}$$

α. ii. Λαμβάνοντας υπόψη τη διαιτησία συμπεραίνουμε ότι στο 20% των περιπτώσεων η μεταφορά κοστίζει 4 κύκλους του σήματος χρονισμού, στο 30% περιπτώσεων κοστίζει 5 κύκλους, στο 40% περιπτώσεων κοστίζει 7 κύκλους, και στο 10% των περιπτώσεων κοστίζει 14 κύκλους.

Επομένως ο μέσος αριθμός κύκλων ρολογιού που απαιτούνται για μια μεταφορά είναι $0,2 \times 4 + 0,3 \times 5 + 0,4 \times 7 + 0,1 \times 14 = 6,5$ κύκλοι και ο μέσος ρυθμός μεταφοράς:

$$(8 \text{ ψηφιολέξεις}) / (6,5 \times 5 \text{ nsec}) = 8 / (6,5 \times 5 \times 10^{-9}) \text{ ψηφιολέξεις / δευτερόλεπτο} = 246,153 \text{ Mψηφιολέξεις / δευτερόλεπτο.}$$

β. Τότε ο μέσος χρόνος που απαιτείται για μια μεταφορά είναι:

$$0,2 \times (12 + 10 + 5) + 0,3 \times (19 + 10 + 5) + 0,4 \times (28 + 10 + 5) + 0,1 \times (62 + 10 + 5) = 40,5 \text{ nsec.}$$

Επομένως ο ρυθμός μεταφοράς είναι: $8 \text{ ψηφιολέξεις} / 40,5 \text{ nsec} = 197,53 \text{ Mψηφιολέξεις / δευτερόλεπτο.}$

Παρατηρούμε ότι η σύγχρονη αρτηρία με δυνατότητα εισαγωγής κύκλων καθυστέρησης επιτυγχάνει το μεγαλύτερο μέσο ρυθμό μεταφοράς πληροφορίας.

Άσκηση 6.3 (6.2 βιβλίου)

Θεωρήστε ένα υπολογιστή με επεξεργαστή συχνότητας λειτουργίας 1 GHz που, εκτός των άλλων, χρησιμοποιείται για να ελέγχει τη λειτουργία 1.000 μονάδων ενός εργοστασίου. Κατά μέσο όρο οι μονάδες εισ/εξ απαιτούν συνολικά εξυπηρέτηση 500 φορές το λεπτό. Να υπολογίσετε το ποσοστό του χρόνου του υπολογιστή που δαπανάται για τον έλεγχο των μονάδων εισ/εξ σε κάθε μια από τις ακόλουθες δύο περιπτώσεις.

α. Ο έλεγχος των μονάδων εισ/εξ βασίζεται στη χρονοπρογραμματισμένη διαδικασία (polling). Για να εξασφαλιστεί η ποιότητα των παραγόμενων προϊόντων θα πρέπει να ελέγχεται η κατάσταση κάθε μονάδας εισ/εξ 10 φορές το δευτερόλεπτο. Υποθέστε ότι για τον έλεγχο της κατάστασης μιας μονάδας εισ/εξ απαιτούνται 500 κύκλοι ρολογιού ενώ στις περιπτώσεις που απαιτείται εξυπηρέτηση, απαιτούνται και άλλοι 500 κύκλοι ρολογιού ανά εξυπηρέτηση.

β. Οι μονάδες εισ/εξ έχουν την δυνατότητα αποστολής και η ΚΜΕ τη δυνατότητα εξυπηρέτησης σημάτων διακοπής. Το κόστος διακοπής της λειτουργίας της ΚΜΕ για την εξυπηρέτηση ενός σήματος διακοπής και η εξυπηρέτηση της διακοπής είναι 1.300 κύκλοι ρολογιού.

γ. Να συγκρίνετε και να δικαιολογήσετε τα αποτελέσματα των περιπτώσεων α και β.

Απάντηση

Αφού η συχνότητα είναι 1 GHz η διάρκεια του κύκλου ρολογιού είναι 1×10^{-9} δευτερόλεπτα = 1 nsec.

α. Αφού πρέπει να ελέγχεται η κατάσταση κάθε μονάδας εισ/εξ 10 φορές το δευτερόλεπτο, κάθε έλεγχος κοστίζει 500 κύκλους ρολογιού και έχουμε 1000 μονάδες εισόδου/εξόδου, η ΚΜΕ κάθε δευτερόλεπτο απασχολείται με τον έλεγχο των μονάδων εισ/εξ για χρόνο ίσο με: $10 \times 500 \times 1.000 \times 1 \times 10^{-9}$ δευτερόλεπτα.

Οι μονάδες εισ/εξ απαιτούν συνολικά 500 φορές εξυπηρέτηση το λεπτό, δηλαδή 500/60 φορές το δευτερόλεπτο. Επομένως οι μονάδες εισ/εξ απαιτούν για την εξυπηρέτησή τους επιπλέον $500/60 \times 500 \times 1 \times 10^{-9}$ δευτερόλεπτα. Επομένως η ΚΜΕ κάθε δευτερόλεπτο απασχολείται με τον έλεγχο και την εξυπηρέτηση των μονάδων εισ/εξ:

$$\begin{aligned} & (10 \times 500 \times 1.000 \times 1 \times 10^{-9}) + (500/60 \times 500 \times 1 \times 10^{-9}) \text{ δευτερόλεπτα} = \\ & = 5.000.000 \times 10^{-9} + 4.166 \times 10^{-9} \text{ δευτερόλεπτα} \cong 5 \times 10^{-3} \text{ δευτερόλεπτα} = \\ & = 0,5 \times 10^{-2} \text{ δευτερόλεπτα.} \end{aligned}$$

Επομένως η ΚΜΕ δαπανά το 0,5% του χρόνου της για τον έλεγχο και την εξυπηρέτηση των μονάδων εισ/εξ.

β. Αφού κατά μέσο όρο όλες οι μονάδες εισ/εξ στέλνουν συνολικά 500 σήματα διακοπής το λεπτό και η εξυπηρέτηση κάθε σήματος διακοπής απαιτεί 1.300 κύκλους ρολογιού, συνεπάγεται ότι στο ένα δευτερόλεπτο η ΚΜΕ αφιερώνει για την εξυπηρέτηση των μονάδων εισ/εξ:

$$\begin{aligned} & 500/60 \times 1.300 \times 1 \times 10^{-9} \text{ δευτερόλεπτα} = 10.833,3 \times 10^{-9} \text{ δευτερόλεπτα} = \\ & = 0,001 \times 10^{-2} \text{ δευτερόλεπτα.} \end{aligned}$$

Επομένως σ' αυτή την περίπτωση η ΚΜΕ δαπανά το 0,001% του χρόνου της για την εξυπηρέτηση των μονάδων εισ/εξ.

γ. Είναι προφανές ότι στη χρονοπρογραμματισμένη διαδικασία η ΚΜΕ δαπανά περισσότερο χρόνο για την εξυπηρέτηση των μονάδων εισ/εξ διότι ελέγχει την κατάστασή τους σε τακτά χρονικά διαστήματα άσχετα αν χρειάζονται ή όχι εξυπηρέτηση. Αντίθετα στην περίπτωση χρήσης σημάτων διακοπής η ΚΜΕ δαπανά χρόνο για την εξυπηρέτηση των μονάδων εισ/εξ μόνο όταν αυτές χρειάζονται πράγματι εξυπηρέτηση.

Άσκηση 6.4 (6.3 βιβλίου)

Θεωρήστε ένα πρόγραμμα κατά την εκτέλεση του οποίου εκτελούνται 5.000.100 εντολές, οι 100 εκ των οποίων είναι εντολές εισόδου/εξόδου δεδομένων από το μαγνητικό δίσκο και είναι διάσπαρτες μέσα στο πρόγραμμα. Ο μέσος αριθμός κύκλων που απαιτούνται για την εκτέλεση κάθε εντολής που δεν είναι εντολή εισόδου/εξόδου είναι 6. Κατά την εκτέλεση μίας εντολής εισόδου/εξόδου μεταφέρονται από το μαγνητικό δίσκο στη κύρια μνήμη 4 Κψηφιολέξεις (KBytes). Έστω ότι το πρόγραμμα εκτελείται σε υπολογιστή που χρησιμοποιεί την τεχνική Άμεσης Προσπέλασης Μνήμης, ΑΠΜ, (Direct Memory Access, DMA). Η διαδικασία αρχικοποίησης του μηχανισμού Άμεσης Προσπέλασης Μνήμης και η ενημέρωση μετά την ολοκλήρωση της μεταφοράς κοστίζουν 2000 κύκλους ρολογιού. Η μεταφορά πληροφορίας μεταξύ της συσκευής του μαγνητικού δίσκου και του καταχωρητή προσωρινής αποθήκευσης του ελεγκτή της συσκευής μαγνη-

τικού δίσκου γίνεται με ρυθμό 4 Μψηφιολέξεις/δευτερόλεπτο (έχει συμπεριληφθεί και ο χρόνος προσπέλασης), ενώ η μεταφορά της πληροφορίας μεταξύ του καταχωρητή προσωρινής αποθήκευσης του ελεγκτή της συσκευής μαγνητικού δίσκου και της κύριας μνήμης γίνεται με ρυθμό 100 Μψηφιολέξεις/δευτερόλεπτο και η συχνότητα λειτουργίας της ΚΜΕ είναι 1 GHz. Θεωρήστε ότι το πρόγραμμα εκτελείται μόνο του στον εν λόγω υπολογιστή και υπολογίστε το μέγιστο πιθανό συνολικό χρόνο ολοκλήρωσής του.

Προσοχή: Η ποσότητα της πληροφορίας μετριέται σε δυνάμεις του 2 ενώ ο ρυθμός μεταφοράς της πληροφορίας σε δυνάμεις του 10.

Απάντηση

Αφού θέλουμε το μέγιστο πιθανό συνολικό χρόνο ολοκλήρωσης του προγράμματος, θεωρούμε ότι και οι 100 εντολές είναι εντολές εισόδου και όταν συναντάται μια από αυτές δεν μπορεί να εκτελεστεί άλλη εντολή του προγράμματος εάν δεν ολοκληρωθεί η διαδικασία εισόδου.

Για την εκτέλεση των 5.000.000 εντολών, οι οποίες δεν είναι εντολές εισόδου, απαιτούνται $(5.000.000 \text{ εντολές}) \times [(6 \text{ κύκλοι ρολογιού}) / \text{εντολή}] \times [(1 \text{ nsec}) / (\text{κύκλος ρολογιού})] = 30.000.000 \text{ nsec} = 30 \text{ msec}$.

Αφού οι εντολές εισόδου είναι διάσπαρτες μέσα στο πρόγραμμα μεταξύ των άλλων εντολών, για την εκτέλεση κάθε εντολής εισόδου απαιτούνται 2.000 κύκλοι ρολογιού για την αρχικοποίηση της διαδικασίας ΑΠΜ και την ενημέρωση της ΚΜΕ μετά την ολοκλήρωση, δηλαδή, $2.000 \times 1 \text{ nsec} = 2 \text{ μsec}$.

Κατά την εκτέλεση μιας εντολής εισόδου μεταφέρεται από τη συσκευή μαγνητικού δίσκου στον καταχωρητή προσωρινής αποθήκευσης 4 Κψηφιολέξεις με ρυθμό 4 Μψηφιολέξεις/δευτερόλεπτο, άρα σε χρόνο:

$T_1 = (4 \text{ Κψηφιολέξεις}) / (4 \text{ Μψηφιολέξεις ανά δευτερόλεπτο}) = (4 \times 2^{10}) / (4 \times 10^6) \text{ δευτερόλεπτα} = 2^{10} / 10^6 \text{ δευτερόλεπτα} = 1024 / 10^6 \text{ δευτερόλεπτα} = 1,024 \text{ msec}$.

Επίσης μεταφέρονται 4 Κψηφιολέξεις από τον καταχωρητή προσωρινής αποθήκευσης στην κύρια μνήμη με ρυθμό 100 Μψηφιολέξεις/δευτερόλεπτο, δηλαδή σε χρόνο:

$T_2 = (4 \text{ Κψηφιολέξεις}) / (100 \text{ Μψηφιολέξεις ανά δευτερόλεπτο}) = (4 \times 2^{10}) / (100 \times 10^6) \text{ δευτερόλεπτα} = 4096 / 10^8 \text{ δευτερόλεπτα} \approx 0,040 \text{ msec}$.

Επομένως για την εκτέλεση μιας εντολής εισόδου απαιτούνται:

$2 \text{ μsec} + 1,024 \text{ msec} + 0,040 \text{ msec} = 0,002 \text{ msec} + 1,024 \text{ msec} + 0,040 \text{ msec} = 1,066 \text{ msec}$.

Λαμβάνοντας υπόψη ότι έχουμε 100 εντολές εισόδου συμπεραίνουμε ότι απαιτείται χρόνος $100 \times 1,066 \text{ msec} = 106,6 \text{ msec}$.

Παρατηρούμε ότι οι 100 εντολές εισόδου κοστίζουν σε χρόνο εκτέλεσης περισσότερο από ότι οι υπόλοιπες 5.000.000 εντολές.

Ο συνολικός χρόνος εκτέλεσης του προγράμματος είναι $30 \text{ msec} + 106,6 \text{ msec} = 136,6 \text{ msec}$.

Άσκηση 6.5

Θεωρήστε ότι στο σύστημα που περιγράφεται στην άσκηση 6.4 η μονάδα μαγνητικού δίσκου αντικαταστάθηκε με άλλη πιο σύγχρονη που διαθέτει και κρυφή μνήμη δίσκου (τώρα η κρυφή μνήμη δίσκου παίζει το ρόλο του καταχωρητή προσωρινής αποθήκευσης του ελεγκτή της συσκευής μαγνητικού δίσκου). Εάν τα υπόλοιπα χαρακτηριστικά της μονάδας του μαγνητικού δίσκου που αναφέρονται στην άσκηση 6.4 παραμένουν τα ίδια και η πιθανότητα η ζητούμενη πληροφορία από τον μαγνητικό δίσκο να βρίσκεται στην κρυφή μνήμη του είναι 0,80 (ρυθμός επιτυχίας, hit ratio) να υπολογίσετε και πάλι το μέγιστο πιθανό χρόνο εκτέλεσης του προγράμματος της άσκησης 6.4.

Απάντηση

Αφού θέλουμε το μέγιστο πιθανό συνολικό χρόνο ολοκλήρωσης του προγράμματος, θεωρούμε ότι και οι 100 εντολές είναι εντολές εισόδου και όταν συναντάται μια από αυτές δεν μπορεί να εκτελεστεί άλλη εντολή του προγράμματος εάν δεν ολοκληρωθεί η διαδικασία εισόδου.

Ο χρόνος εκτέλεσης των 5.000.000 εντολών που δεν είναι εντολές εισόδου δεν επηρεάζεται. Επομένως απαιτούνται $(5.000.000 \text{ εντολές}) \times [(6 \text{ κύκλοι ρολογιού})/\text{εντολή}] \times [(1 \text{ nsec})/\text{κύκλος ρολογιού}] = 30.000.000 \text{ nsec} = 30 \text{ msec}$.

Αφού οι εντολές εισόδου είναι διάσπαρτες μέσα στο πρόγραμμα μεταξύ των άλλων εντολών, για την εκτέλεση κάθε εντολής εισόδου απαιτούνται 2.000 κύκλοι ρολογιού για την αρχικοποίηση της διαδικασίας ΑΠΜ και την ενημέρωση της ΚΜΕ μετά την ολοκλήρωση, δηλαδή, $2.000 \times 1 \text{ nsec} = 2.000 \times 10^{-9} \text{ sec} = 2 \text{ msec}$.

Αφού μόνο στο 20% των περιπτώσεων η ζητούμενη πληροφορία δε βρίσκεται στην κρυφή μνήμη του δίσκου, μόνο σ' αυτές τις περιπτώσεις θα μεταφερθεί από το μαγνητικό δίσκο στην κρυφή μνήμη δίσκου και στη συνέχεια στην κύρια μνήμη. Επομένως ο χρόνος εκτέλεσης των 100 εντολών εισόδου/εξόδου θα είναι:

$$\begin{aligned}
 T &= 100 \times [0,8 \times (4 \text{ Kψηφιολέξεις}) / (100 \text{ Mψηφιολέξεις ανά δευτερόλεπτο}) + \\
 &\quad (1-0,8) \times ((4 \text{ Kψηφιολέξεις}) / (4 \text{ Mψηφιολέξεις ανά δευτερόλεπτο}) + \\
 &\quad (4 \text{ Kψηφιολέξεις}) / (100 \text{ Mψηφιολέξεις ανά δευτερόλεπτο}))] = \\
 &= 100 \times [(4 \text{ Kψηφιολέξεις}) / (100 \text{ Mψηφιολέξεις ανά δευτερόλεπτο}) + \\
 &\quad 0,2 \times (4 \text{ Kψηφιολέξεις}) / (4 \text{ Mψηφιολέξεις ανά δευτερόλεπτο})] = \\
 &= 100 \times [(4 \times 2^{10}) / (100 \times 10^6) \text{ δευτερόλεπτα} + 0,2 \times (4 \times 2^{10}) / (4 \times 10^6) \text{ δευτερόλεπτα}] = \\
 &= 100 \times [0,040 \text{ msec} + 0,2 \times (1,024 \text{ msec})] = \\
 &= 100 \times [0,040 \text{ msec} + 0,20 \text{ msec}] = 100 \times 0,24 \text{ msec} = 24 \text{ msec}.
 \end{aligned}$$

Παρατηρούμε ότι ο χρόνος εκτέλεσης των εντολών εισόδου έχει μειωθεί σε λιγότερο από το ένα τέταρτο του χρόνου που απαιτούνταν στο σύστημα της άσκησης 6.4.

Ο συνολικός χρόνος εκτέλεσης του προγράμματος είναι $30 \text{ msec} + 2 \text{ msec} + 24 \text{ msec} \cong 54 \text{ msec}$.

Άσκηση 6.6

Θεωρήστε μια σύγχρονη αρτηρία επεξεργαστή-μνήμης με συχνότητα λειτουργίας 100 MHz και αρτηρία δεδομένων των 64 δυαδικών ψηφίων. Η διαιτησία της αρτηρίας κοστίζει έναν κύκλο ρολογιού. Επίσης η αποστολή μιας εντολής για μεταφορά δεδομένων από την κύρια μνήμη στην κρυφή μνήμη κοστίζει έναν κύκλο ρολογιού. Ο χρόνος προσπέλασης της κύριας μνήμης είναι 60 nsec. Υποθέστε ότι η κυρία μνήμη έχει οργάνωση 8-δρόμων χαμηλής τάξης διαφύλλωσης με μία ψηφιολέξη ανά θέση μνήμης και ότι έχει υλοποιηθεί με ολοκληρωμένα κυκλώματα μνήμης DRAM τα οποία έχουν τη δυνατότητα γρήγορης προσπέλασης ομάδας θέσεων που έχουν την ίδια διεύθυνση γραμμής (row address) και κάθε γραμμή είναι των 512 ψηφιολέξεων. Για καθεμία των κάτωθι περιπτώσεων να υπολογίσετε το χρόνο που απαιτείται για τη μεταφορά ενός μπλοκ πληροφορίας των 64 ψηφιολέξεων από την κυρία μνήμη σ' ένα πλαίσιο της κρυφής μνήμης. Να υπολογίσετε επίσης το μέγιστο ρυθμό μεταφοράς πληροφορίας στην αρτηρία επεξεργαστή-μνήμης, υποθέτοντας ότι η διαιτησία λαμβάνει χώρα τον επόμενο κύκλο της αρτηρίας μετά την ολοκλήρωση μιας διαδικασίας ανάγνωσης.

α. Η αρτηρία υποστηρίζει ομαδική μεταφορά δεδομένων, μία μεταφορά ανά κύκλο, σε έως και 8 διαδοχικούς κύκλους ρολογιού.

β. Η αρτηρία υποστηρίζει ομαδική μεταφορά δεδομένων, μία μεταφορά ανά κύκλο, σε έως και 4 διαδοχικούς κύκλους ρολογιού.

γ. Η αρτηρία δεν υποστηρίζει ομαδική μεταφορά δεδομένων.

Απάντηση

Εφόσον η συχνότητα της σύγχρονης αρτηρίας είναι 100 MHz, η περίοδος του ρολογιού θα είναι $1/(100 \text{ MHz}) = 1/(100 \times 10^6 \text{ κύκλοι ανά sec}) = 10^{-8} \text{ sec} = 10 \text{ ns}$.

Ένα μπλοκ της κυρίας μνήμης το οποίο μεταφέρεται σε ένα πλαίσιο της κρυφής μνήμης αποτελείται από 64 ψηφιολέξεις, δηλαδή από οκτώ ομάδες των οκτώ ψηφιολέξεων. Επειδή η κυρία μνήμη έχει υλοποιηθεί με ολοκληρωμένα κυκλώματα μνήμης DRAM τα οποία έχουν τη δυνατότητα γρήγορης προσπέλασης ομάδας θέσεων που έχουν την ίδια διεύθυνση γραμμής (row address) και κάθε γραμμή είναι των 512 ψηφιολέξεων, οι 64 ψηφιολέξεις που θέλουμε να προσπελάσουμε έχουν πάντα την ίδια διεύθυνση γραμμής. Επομένως, ο χρόνος προσπέλασης της κύριας μνήμης, ο οποίος είναι 60 nsec, θα ληφθεί υπ' όψη μόνο για την προσπέλαση της πρώτης ομάδας των οκτώ ψηφιολέξεων. Οποιαδήποτε από τις επόμενες επτά προσπελάσεις κοστίζει ένα κύκλο ρολογιού.

α.

Λαμβάνοντας υπόψη ότι η αρτηρία δεδομένων είναι των 64 δυαδικών ψηφίων και υποστηρίζει την ομαδική μεταφορά δεδομένων σε έως και οκτώ διαδοχικούς κύκλους ρολογιού, συμπεραίνουμε ότι ο χρόνος μεταφοράς T_1 των 64 ψηφιολέξεων θα ισούται με:

$$\begin{aligned} T_1 &= 1 \text{ κύκλος για τη διαιτησία} + 1 \text{ κύκλος για την αποστολή της εντολής} + 60 \text{ ns για την} \\ &\quad \text{προσπέλαση των πρώτων οκτώ ψηφιολέξεων του μπλοκ} + 7 \text{ κύκλοι για την προσπέ-} \\ &\quad \text{λαση των επόμενων επτά ομάδων των οκτώ ψηφιολέξεων του μπλοκ} = \\ &= 10 \text{ ns} + 10 \text{ ns} + 60 \text{ ns} + 7 \times 10 \text{ ns} = 150 \text{ ns} \end{aligned}$$

Ο μέγιστος ρυθμός μεταφοράς M_1 πληροφορίας στην αρτηρία επεξεργαστή-μνήμης ισούται με:

$$\begin{aligned} M_1 &= (64 \text{ ψηφιολέξεις}) / (150 \text{ ns}) = 0,42666 \times 10^9 \text{ ψηφιολέξεις ανά δευτερόλεπτο} = \\ &= 426,66 \times 10^6 \text{ ψηφιολέξεις ανά δευτερόλεπτο} = \\ &= 426,66 \text{ Mψηφιολέξεις ανά δευτερόλεπτο} \end{aligned}$$

β.

Σ' αυτή την περίπτωση, η αρτηρία υποστηρίζει ομαδική μεταφορά δεδομένων σε έως και 4 διαδοχικούς κύκλους ρολογιού. Επομένως μετά τη μεταφορά των τεσσάρων ομάδων, των οκτώ ψηφιολέξεων η καθεμία, θα πρέπει η διαιτησία να χορηγήσει και πάλι την αρτηρία και να σταλεί μια εντολή για τη μεταφορά δεδομένων από την κύρια μνήμη στην κρυφή μνήμη. Επομένως ο χρόνος μεταφοράς T_2 των 64 ψηφιολέξεων θα ισούται με:

$$\begin{aligned}
 T_1 &= 1 \text{ κύκλος για τη διαιτησία} + 1 \text{ κύκλος για την αποστολή της εντολής} + 60 \text{ ns για την} \\
 &\text{προσπέλαση των πρώτων οκτώ ψηφιολέξεων του μπλοκ} + 3 \text{ κύκλοι για την προσπέ-} \\
 &\text{λαση των επόμενων τριών ομάδων των οκτώ ψηφιολέξεων του μπλοκ} + \\
 &1 \text{ κύκλος για τη διαιτησία} + 1 \text{ κύκλος για την αποστολή της εντολής} + 4 \text{ κύκλοι για} \\
 &\text{την προσπέλαση των επόμενων τεσσάρων ομάδων των οκτώ ψηφιολέξεων του} \\
 &\text{μπλοκ} = \\
 &= 10 \text{ ns} + 10 \text{ ns} + 60 \text{ ns} + 3 \times 10 \text{ ns} + 10 \text{ ns} + 10 \text{ ns} + 4 \times 10 \text{ ns} = 170 \text{ ns}
 \end{aligned}$$

Ο μέγιστος ρυθμός μεταφοράς M_2 πληροφορίας στην αρτηρία επεξεργαστή-μνήμης ισούται με:

$$\begin{aligned}
 M_2 &= (64 \text{ ψηφιολέξεις}) / (170 \text{ ns}) = 0,37647 \times 10^9 \text{ ψηφιολέξεις ανά δευτερόλεπτο} = \\
 &= 376,47 \times 10^6 \text{ ψηφιολέξεις ανά δευτερόλεπτο} = \\
 &= 376,47 \text{ Mψηφιολέξεις ανά δευτερόλεπτο.}
 \end{aligned}$$

γ.

Όταν η αρτηρία δεν υποστηρίζει ομαδική μεταφορά δεδομένων, τότε μετά τη μεταφορά κάθε ομάδας των 8 ψηφιολέξεων απαιτείται η διαιτησία να χορηγήσει και πάλι την αρτηρία και να σταλεί μια εντολή για τη μεταφορά δεδομένων από την κύρια μνήμη στην κρυφή μνήμη. Επομένως ο χρόνος μεταφοράς T_3 των 64 ψηφιολέξεων θα ισούται με:

$$\begin{aligned}
 T_3 &= 1 \text{ κύκλος για τη διαιτησία} + 1 \text{ κύκλος για την αποστολή της εντολής} + 60 \text{ ns για την} \\
 &\text{προσπέλαση των πρώτων οκτώ ψηφιολέξεων του μπλοκ} + 7 \times (1 \text{ κύκλος για τη διαι-} \\
 &\text{τησία} + 1 \text{ κύκλος για την αποστολή της εντολής} + \text{ένας κύκλος για την προσπέλαση} \\
 &\text{μιας ομάδας των οκτώ ψηφιολέξεων}) = \\
 &= 10 \text{ ns} + 10 \text{ ns} + 60 \text{ ns} + 7 \times (10 \text{ ns} + 10 \text{ ns} + 10 \text{ ns}) = 290 \text{ ns}
 \end{aligned}$$

Ο μέγιστος ρυθμός μεταφοράς M_3 πληροφορίας στην αρτηρία επεξεργαστή-μνήμης ισούται με:

$$\begin{aligned}
 M_3 &= (64 \text{ ψηφιολέξεις}) / (290 \text{ ns}) = 0,22068 \times 10^9 \text{ ψηφιολέξεις ανά δευτερόλεπτο} = \\
 &= 220,68 \times 10^6 \text{ ψηφιολέξεις ανά δευτερόλεπτο} = \\
 &= 220,68 \text{ Mψηφιολέξεις ανά δευτερόλεπτο.}
 \end{aligned}$$

Άσκηση 6.7 (6.4 βιβλίου)

Θεωρήστε ένα υπολογιστή με συχνότητα λειτουργίας 200 MHz και μία συσκευή εισόδου που εισάγει δεδομένα σε μπλοκ 1 Mψηφιολέξεων (MBytes). Ο ρυθμός μεταφοράς από τη συσκευή εισόδου στον καταχωρητή του ελεγκτή της είναι 8 Kψηφιολέξεις το δευτερόλεπτο. Τα δεδομένα μεταφέρονται από τη μονάδα εισόδου προς τη μνήμη σε ομάδες των 4 ψηφιολέξεων.

Όταν χρησιμοποιείται η προγραμματισμένη διαδικασία εισόδου-εξόδου με χρήση σημάτων διακοπής, το κόστος κάθε διακοπής είναι 500 χρονικές περιόδους (cycles) ενώ το κόστος μεταφοράς 4 ψηφιολέξεων από τη μονάδα εισόδου στην κύρια μνήμη είναι 80 χρονικές περιόδους.

i. Διερευνήστε σε ποια από τις ακόλουθες δύο τεχνικές απασχολείται λιγότερο η κεντρική μονάδα επεξεργασίας (ΚΜΕ):

α. Ο επεξεργαστής δέχεται ένα σήμα διακοπής (interrupt) για τη μεταφορά κάθε 4 ψηφιολέξεων ή

β. Ο επεξεργαστής δέχεται ένα σήμα διακοπής στην αρχή της μεταφοράς κάθε μπλοκ δεδομένων από τη μονάδα εισόδου;

ii. Ποια σχέση πρέπει να ισχύει όσον αφορά το ρυθμό εισαγωγής δεδομένων της μονάδας εισόδου, ώστε η κεντρική μονάδα επεξεργασίας να απασχολείται για λιγότερες χρονικές περιόδους όταν δεν υλοποιηθεί η τεχνική που σας προέκυψε από τα τρέχοντα δεδομένα αλλά η άλλη τεχνική;

Παρατήρηση: Όταν αναφερόμαστε σε μεταφορά πληροφορίας ισχύει ότι 1K ψηφιολέξεις = 1.000 ψηφιολέξεις και 1Mψηφιολέξεις = 1.000.000 ψηφιολέξεις

Απάντηση

Αφού η συχνότητα είναι 200 MHz, η διάρκεια του κύκλου ρολογιού είναι $1/(200 \times 10^6)$ δευτερόλεπτα = $0,5 \times 10^{-8}$ sec = 5×10^{-9} sec = 5 nsec.

Ας υποθέσουμε ότι από τις 500 χρονικές περιόδους που είναι το κόστος κάθε διακοπής οι μισές, δηλαδή οι 250 χρονικές περιόδους, καταναλίσκονται για την αποθήκευση του πτητικού περιβάλλοντος της ΚΜΕ πριν την εξυπηρέτηση του σήματος διακοπής και οι υπόλοιπες 250 χρονικές περιόδους για την αποκατάσταση του πτητικού περιβάλλοντος της ΚΜΕ.

Αφού ο ρυθμός μεταφοράς από τη μονάδα εισόδου στον καταχωρητή του ελεγκτή της μονάδας εισόδου είναι 8 Kψηφιολέξεις το δευτερόλεπτο, για την μεταφορά 4 ψηφιολέξεων απαιτούνται:

$$T_{4\psi} = (4 \text{ ψηφιολέξεις}) / (8 \text{ Kψηφιολέξεις ανά δευτερόλεπτο}) = 0,5 \times 10^{-3} \text{ δευτερόλεπτα} = 500.000 \text{ nsec}$$

α.

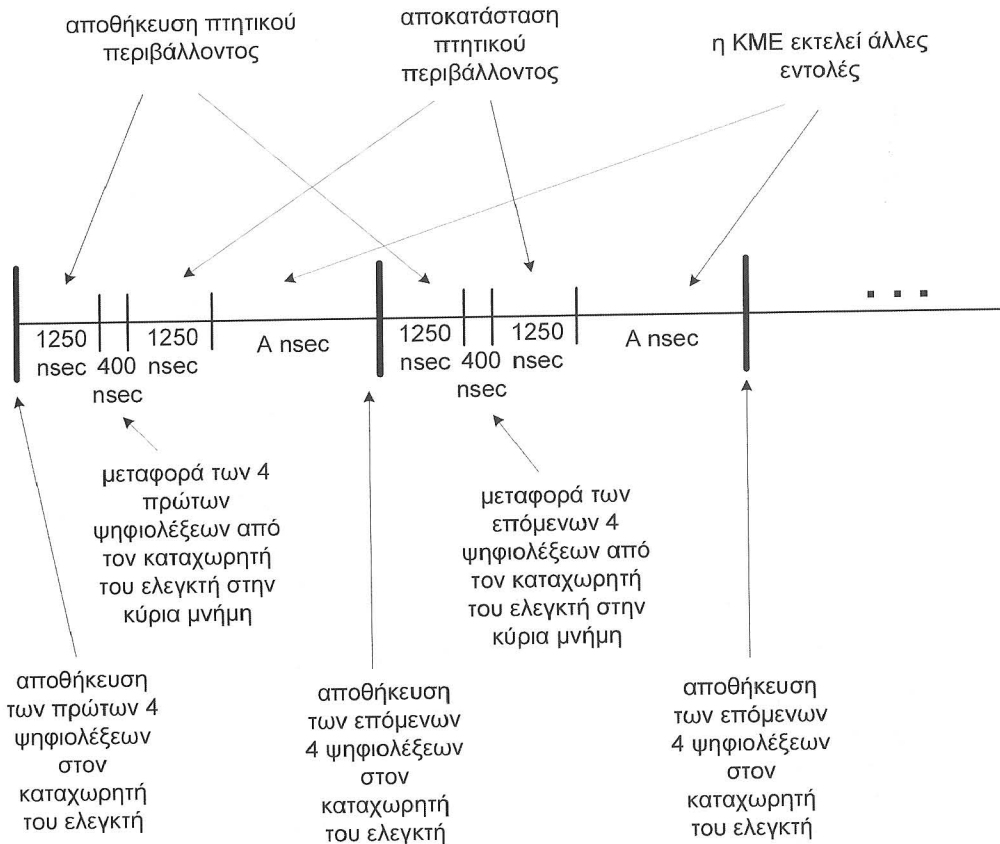
Κάθε φορά που 4 ψηφιολέξεις είναι διαθέσιμες στον καταχωρητή της μονάδας εισόδου, η ΚΜΕ δέχεται ένα σήμα διακοπής. Το κόστος κάθε διακοπής είναι 500 χρονικές περίοδοι (κύκλοι ρολογιού) ενώ για τη μεταφορά των 4 ψηφιολέξεων από τη μονάδα εισόδου στην κύρια μνήμη απαιτούνται 80 χρονικές περίοδοι (βλέπε σχήμα 6.7.1). Επομένως το συνολικό κόστος για την εξυπηρέτηση του σήματος διακοπής είναι:

$$T_{4\psi,\sigma\delta} = (500+80) \text{ χρονικές περίοδοι} = 580 \times 5 \text{ nsec} = 2900 \text{ nsec}$$

Έχουμε υποθέσει ότι από τις 500 χρονικές περιόδους, που είναι το κόστος κάθε διακοπής, οι μισές, δηλαδή οι 250 χρονικές περίοδοι, καταναλίσκονται μετά την μεταφορά των 4 ψηφιολέξεων στην κύρια μνήμη, για την αποκατάσταση του πτητικού περιβάλλοντος της ΚΜΕ. Επειδή η χρονική διάρκεια των 250 χρονικών περιόδων είναι $250 \times 5 \text{ nsec} = 1250 \text{ nsec}$, και ο χρόνος εισαγωγής 4 ψηφιολέξεων από τη μονάδα εισόδου στον καταχωρητή της μονάδας ελέγχου της είναι 500.000 nsec, μέχρι να δεχθεί η ΚΜΕ το επόμενο σήμα διακοπής θα έχει στη διάθεσή της για την εκτέλεση άλλων εντολών $500.000 \text{ nsec} - 1250 \text{ nsec} = 498.750 \text{ nsec} = 99.750 \text{ χρονικές περιόδους}$.

Ο συνολικός χρόνος τον οποίο δαπανά η ΚΜΕ για τη μεταφορά ενός μπλοκ 1 Μψηφιολέξεων είναι:

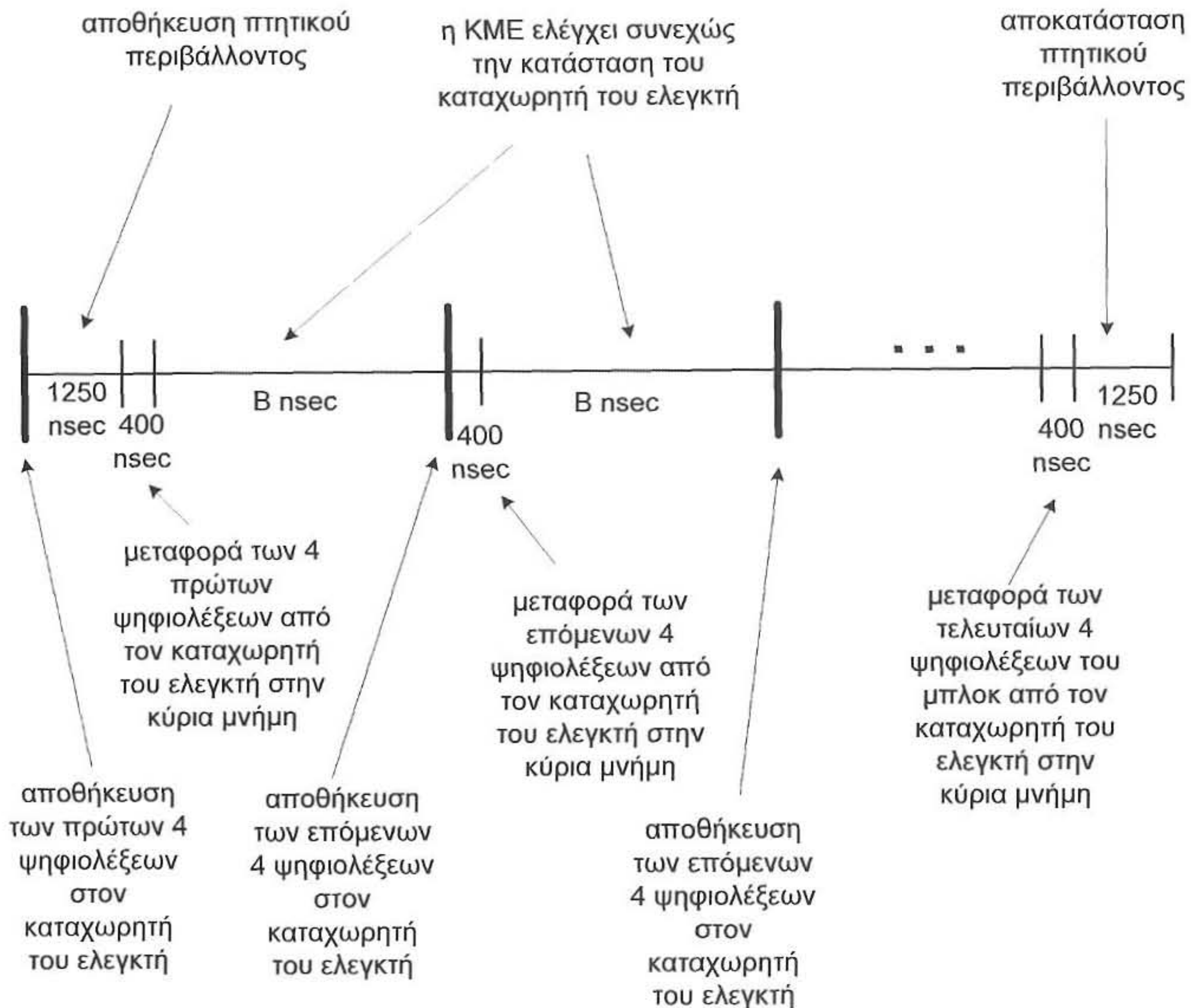
$$\begin{aligned} T_{1M\psi,\alpha} &= (\text{πλήθος ομάδων των 4 ψηφιολέξεων}) \times [(\text{κόστος σήματος διακοπής}) + \\ &\quad (\text{χρόνος μεταφοράς 4 ψηφιολέξεων από τον καταχωρητή του ελεγκτή της μονάδας εισόδου στην κύρια μνήμη})] = \\ &= (1M\psi\text{ψηφιολέξεις}/4) \times [(500 \text{ χρονικές περίοδοι}) + (80 \text{ χρονικές περίοδοι})] \times 5 \text{ nsec} \\ &= 262.144 \times 580 \times 5 \text{ nsec} = 760.217.600 \text{ nsec} \cong 760,217 \text{ msec.} \end{aligned}$$



Σχήμα 6.7.1 Ο επεξεργαστής δέχεται ένα σήμα διακοπής (interrupt) για τη μεταφορά κάθε 4 ψηφιολέξεων.

β.

Σ' αυτή την περίπτωση η ΚΜΕ θα δεχτεί ένα σήμα διακοπής μόνο όταν οι πρώτες 4 ψηφιολέξεις του μπλοκ των 1 Μψηφιολέξεων έχουν μεταφερθεί από τη μονάδα εισόδου στον καταχωρητή του ελεγκτή της. Τότε θα δαπανήσει 250 χρονικές περιόδους για την αποθήκευση του πτητικού της περιβάλλοντος και στη συνέχεια 80 χρονικές περιόδους για την μεταφορά των 4 ψηφιολέξεων στην κύρια μνήμη. Μετά την ολοκλήρωση της μεταφοράς θα πρέπει να ελέγχει συνεχώς την κατάσταση του ελεγκτή της μονάδας εισόδου για να διαπιστώσει πότε οι επόμενες 4 ψηφιολέξεις είναι διαθέσιμες για να τις μεταφέρει στην κύρια μνήμη (βλέπε σχήμα 6.7.2).



Σχήμα 6.7.2 Ο επεξεργαστής δέχεται ένα σήμα διακοπής στην αρχή της μεταφοράς κάθε μπλοκ δεδομένων από τη μονάδα εισόδου.

Αφού η ΚΜΕ για να μεταφέρει 4 ψηφιολέξεις από τον καταχωρητή του ελεγκτή της μονάδας εισόδου στην κύρια μνήμη δαπανά 80 χρονικές περιόδους, συμπεραίνουμε ότι για τον έλεγχο της κατάστασης του ελεγκτή της μονάδας εισόδου η ΚΜΕ θα δαπανά πολύ μικρότερο αριθμό χρονικών περιόδων, έστω τ_k . Επειδή $\tau_k \times 5 \text{ nsec}$ είναι πολύ μικρότερο των 500.000 nsec που απαιτούνται για τη μεταφορά 4 ψηφιολέξεων από τη μονάδα εισόδου στον καταχωρητή του ελεγκτή της, θεωρούμε ότι η διαδικασία ελέγχου της κατάστασης του ελεγκτή της μονάδας εισόδου από την ΚΜΕ δεν βάζει καμιά καθυστέρηση.

Ο συνολικός χρόνος τον οποίο δαπανά η ΚΜΕ για τη μεταφορά ενός μπλοκ 1 Μψηφιολέξεων είναι:

$$\begin{aligned}
 T_{1M\psi\beta} &= [(\text{χρόνος αποθήκευσης πτητικού περιβάλλοντος λόγω του σήματος διακοπής}) + \\
 &\quad (\text{χρόνος μεταφοράς 4 ψηφιολέξεων από τον καταχωρητή του ελεγκτή της μονάδας εισόδου στην κύρια μνήμη})] + (\text{πλήθος ομάδων των 4 ψηφιολέξεων} - 1) \times \\
 &\quad [(\text{κόστος μεταφοράς 4 ψηφιολέξεων από τη μονάδα εισόδου στον καταχωρητή του ελεγκτή της}) + (\text{χρόνος μεταφοράς από τον καταχωρητή του ελεγκτή της μονάδας εισόδου στην κύρια μνήμη})] + (\text{χρόνος αποκατάστασης του πτητικού περιβάλλοντος της ΚΜΕ}) = \\
 &= 250 \times 5 \text{ nsec} + 80 \times 5 \text{ nsec} + (1M\text{ψηφιολέξεις}/4 - 1) \times (500.000 \text{ nsec} + 80 \times 5 \text{ nsec}) + \\
 &\quad 250 \times 5 \text{ nsec} = \\
 &= 1.250 \text{ nsec} + 400 \text{ nsec} + 262.143 \times (500.000 \text{ nsec} + 400 \text{ nsec}) + 1.250 \text{ nsec} = \\
 &= 1.650 \text{ nsec} + 131.176,3572 \times 10^6 \text{ nsec} + 1.250 \text{ nsec} = \\
 &= 0,001650 \times 10^6 \text{ nsec} + 131.176,3572 \times 10^6 \text{ nsec} + 0,001250 \times 10^6 \text{ nsec} = \\
 &= (0,001650 + 131.176,3572 + 0,001250) \times 10^6 \text{ nsec} = \\
 &= 131.176,3601 \times 10^6 \text{ nsec} \\
 &= 131.176,3601 \text{ msec}
 \end{aligned}$$

Παρατηρούμε ότι σ' αυτή την περίπτωση ο χρόνος που δαπανά η ΚΜΕ για την διαδικασία εισόδου είναι πολύ μεγαλύτερος.

γ.

Από το σχήμα 6.7.1 παρατηρούμε ότι, όταν υλοποιείται η πρώτη τεχνική, μεταξύ δύο διαδοχικών μεταφορών πληροφορίας από τον ελεγκτή της μονάδας εισόδου στην κύρια μνήμη η ΚΜΕ δαπανά $2 \times 250 \times 5 \text{ nsec}$. Όταν υλοποιείται η δεύτερη τεχνική, τότε ο χρόνος μεταξύ δύο διαδοχικών μεταφορών πληροφορίας από τον ελεγκτή της μονάδας εισόδου στην κύρια μνήμη ισούται με τον χρόνο μεταφοράς 4 ψηφιολέξεων από την μονάδα εισόδου στον καταχωρητή του ελεγκτή της. Αυτός ο χρόνος ισούται με το χρόνο που δαπανά η ΚΜΕ για τον έλεγχο της κατάστασης του καταχωρητή δεδομένων του ελεγκτή της μονάδας εισόδου. Επομένως η δεύτερη μέθοδος θα είναι καλύτερη της πρώτης όταν ο χρόνος που απαιτείται για τη μεταφορά 4 ψηφιολέξεων από τη μονάδα εισόδου στον καταχωρητή του ελεγκτή της είναι μικρότερος του $2 \times 250 \times 5 \text{ nsec}$. Δηλαδή όταν

$$4/(\text{ρυθμός μεταφοράς από την μονάδα εισόδου στον καταχωρητή του ελεγκτή της}) < 2 \times 250 \times 5 \text{ nsec} \text{ ή}$$

$$4/(2 \times 250 \times 5 \text{ nsec}) < (\text{ρυθμός μεταφοράς από την μονάδα εισόδου στον καταχωρητή του ελεγκτή της}) \text{ ή}$$

(ρυθμός μεταφοράς από την μονάδα εισόδου στον καταχωρητή του ελεγκτή της) >
 $(4 \times 10^9)/(2500)$ ψηφιολέξεις ανά δευτερόλεπτο =
 1.600.000 ψηφιολέξεις ανά δευτερόλεπτο = 1,6 Mψηφιολέξεις το δευτερόλεπτο.

Προς επιβεβαίωση επαναυπολογίζουμε την τιμή του $T_{IM\psi,\beta}$ για ρυθμό μεταφοράς από την μονάδα εισόδου στον καταχωρητή του ελεγκτή της, PM, ίσο με 1,61 Mψηφιολέξεις το δευτερόλεπτο. Η τιμή του $T_{IM\psi,\alpha}$ δεν επηρεάζεται από την τιμή του PM.

Για τη μεταφορά 4 ψηφιολέξεων από τη μονάδα εισόδου στον καταχωρητή του ελεγκτή της μονάδας εισόδου απαιτούνται:

$$T_{4\psi} = (4 \text{ ψηφιολέξεις}) / (1,61 \text{ Mψηφιολέξεις ανά δευτερόλεπτο}) \cong 2.484 \text{ nsec}$$

Επομένως

$$\begin{aligned} T_{IM\psi,\beta} &= [(\text{χρόνος αποθήκευσης πτητικού περιβάλλοντος λόγω του σήματος διακοπής}) + \\ &\quad (\text{χρόνος μεταφοράς 4 ψηφιολέξεων από τον καταχωρητή του ελεγκτή της μονά-} \\ &\quad \text{δας εισόδου στην κύρια μνήμη})] + (\text{πλήθος ομάδων των 4 ψηφιολέξεων} - 1) \times \\ &\quad [(\text{κόστος μεταφοράς 4 ψηφιολέξεων από τη μονάδα εισόδου στον καταχωρητή} \\ &\quad \text{του ελεγκτή της}) + (\text{χρόνος μεταφοράς από τον καταχωρητή του ελεγκτή της} \\ &\quad \text{μονάδας εισόδου στην κύρια μνήμη})] + (\text{χρόνος αποκατάστασης του πτητικού} \\ &\quad \text{περιβάλλοντος της KME}) = \\ &= 250 \times 5 \text{ nsec} + 80 \times 5 \text{ nsec} + (1 \text{ Mψηφιολέξεις} / 4 - 1) \times [2.484 \text{ nsec} + 80 \times 5 \text{ nsec}] + \\ &\quad 250 \times 5 \text{ nsec} = \\ &= 1250 \text{ nsec} + 400 \text{ nsec} + 262.143 \times [2.484 \text{ nsec} + 400 \text{ nsec}] + 1250 \text{ nsec} = \\ &= 756.023.312 \text{ nsec} \cong 756,023 \text{ msec.} \end{aligned}$$

Παρατηρούμε ότι για $PM = 1,61 \text{ Mψηφιολέξεις το δευτερόλεπτο} > 1,6 \text{ Mψηφιολέξεις το δευτερόλεπτο}$, η δεύτερη τεχνική οδηγεί σε πιο αποδοτική λύση όσον αφορά το χρόνο που δαπανά η KME.

Άσκηση 6.8

Θεωρήστε ένα μαγνητικό δίσκο με τμήματα (sectors) των 512 ψηφιολέξεων, 1024 τμήματα ανά ομόκεντρο κύκλο και μέσο χρόνο προσπέλασης 5 ms. Σχεδιάστε το ρυθμό μεταφοράς δεδομένων εισόδου/εξόδου καθώς το μέγεθος της μονάδας πληροφορίας που μεταφέρεται μεταβάλλεται σε μέγεθος από 1 τμήμα μέχρι 1024 διαδοχικά τμήματα στον ίδιο ομόκεντρο κύκλο, για κάθε μια των περιπτώσεων:

- Ο μέγιστος ρυθμός μεταφοράς είναι 5 Mψηφιολέξεις το δευτερόλεπτο και
- Ο μέγιστος ρυθμός μεταφοράς 20 Mψηφιολέξεις το δευτερόλεπτο.

Αγνοήστε οποιαδήποτε άλλη καθυστέρηση. Να σχολιάσετε τις καμπύλες που προκύπτουν.

Απάντηση

Υπενθυμίζουμε ότι ο μέσος χρόνος προσπέλασης ισούται με το άθροισμα του μέσου χρόνου αναζήτησης και του μέσου χρόνου αναμονής. Μέγιστος ρυθμός μεταφοράς είναι ο ρυθμός ανάγνωσης από την επιφάνεια του δίσκου.

Όταν σε κάθε προσπέλαση του μαγνητικού δίσκου μεταφέρεται το περιεχόμενο κ τμημάτων του και ο μέγιστος ρυθμός μεταφοράς είναι MPM, τότε ο συνολικός χρόνος για τη μεταφορά των κ τμημάτων είναι:

$$\begin{aligned} T_k &= (\text{μέσος χρόνος προσπέλασης}) + \\ &\quad (\text{ποσότητα που μεταφέρεται κάθε φορά})/(\text{μέγιστος ρυθμός μεταφοράς}) = \\ &= 5 \text{ ms} + (k \times 512 \text{ ψηφιολέξεις})/\text{MPM} \end{aligned}$$

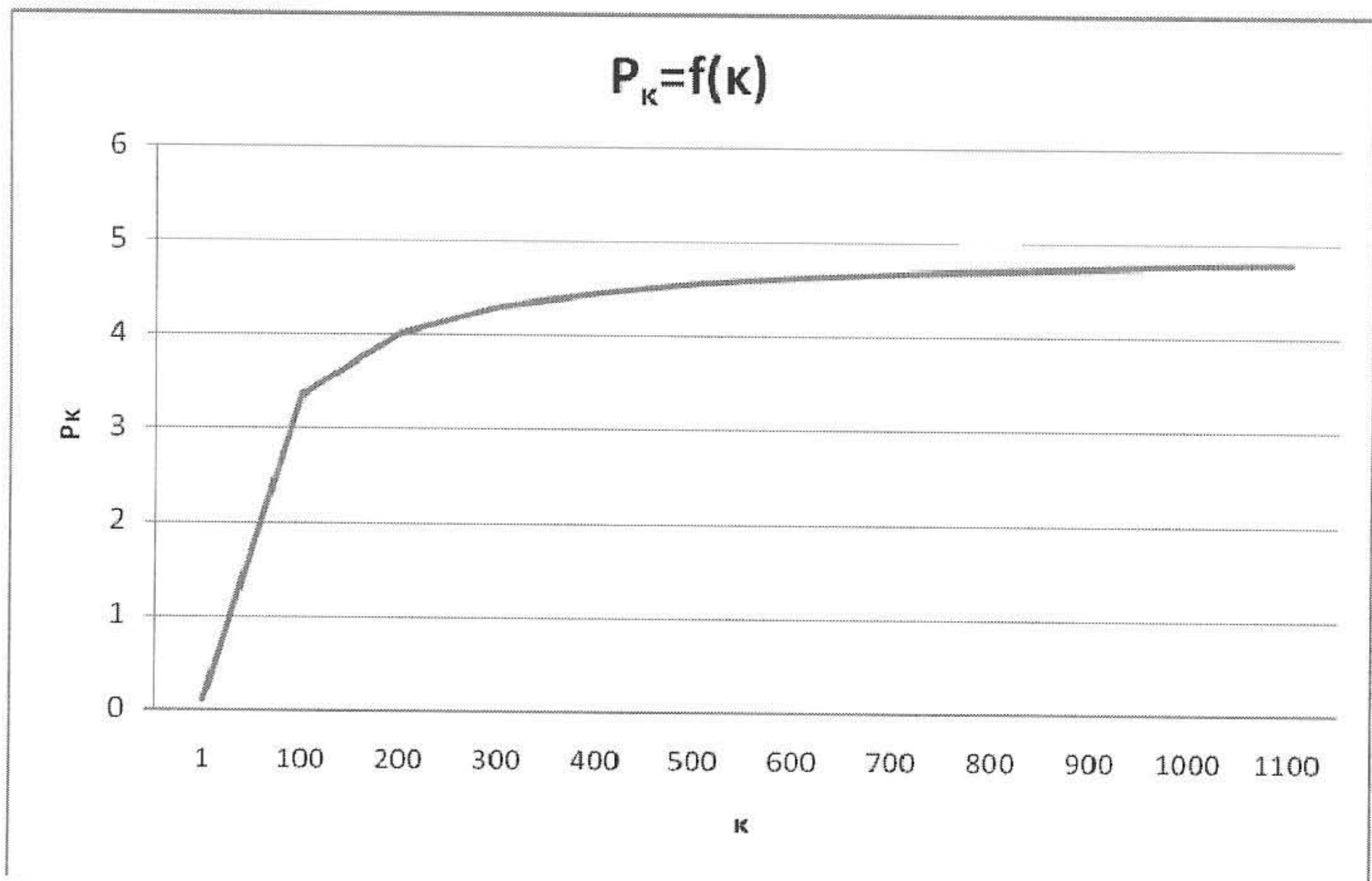
Ο ρυθμός μεταφοράς δεδομένων εισόδου/εξόδου δίνεται από τη σχέση:

$$\begin{aligned} P_k &= (\text{ποσότητα που μεταφέρεται κάθε φορά})/(\text{συνολικός χρόνος για τη μεταφορά των κ τμημάτων}) = \\ &= (k \times 512 \text{ ψηφιολέξεις})/T_k = \\ &= (k \times 512 \text{ ψηφιολέξεις})/(5 \text{ ms} + (k \times 512 \text{ ψηφιολέξεις})/\text{MPM}) \\ &= 1/((5 \text{ ms})/(k \times 512 \text{ ψηφιολέξεις}) + 1/\text{MPM}) \quad (6.8.1) \end{aligned}$$

Από την ανωτέρω σχέση προκύπτει ότι όσο αυξάνεται η τιμή του κ (το πλήθος των τμημάτων που μεταφέρονται ως μονάδα από ή προς τη μονάδα του δίσκου) τόσο μικραίνει η τιμή του όρου $(5 \text{ ms})/(k \times 512 \text{ ψηφιολέξεις})$, οπότε αυξάνεται η τιμή του ρυθμού μεταφοράς δεδομένων εισόδου/εξόδου P_k . Όταν η τιμή του κ τείνει στο άπειρο, η τιμή του όρου $(5 \text{ ms})/(k \times 512 \text{ ψηφιολέξεις})$ τείνει στο μηδέν, οπότε η τιμή του P_k τείνει στην τιμή του MPM.

α. Ο μέγιστος ρυθμός μεταφοράς είναι 5 Mψηφιολέξεις το δευτερόλεπτο

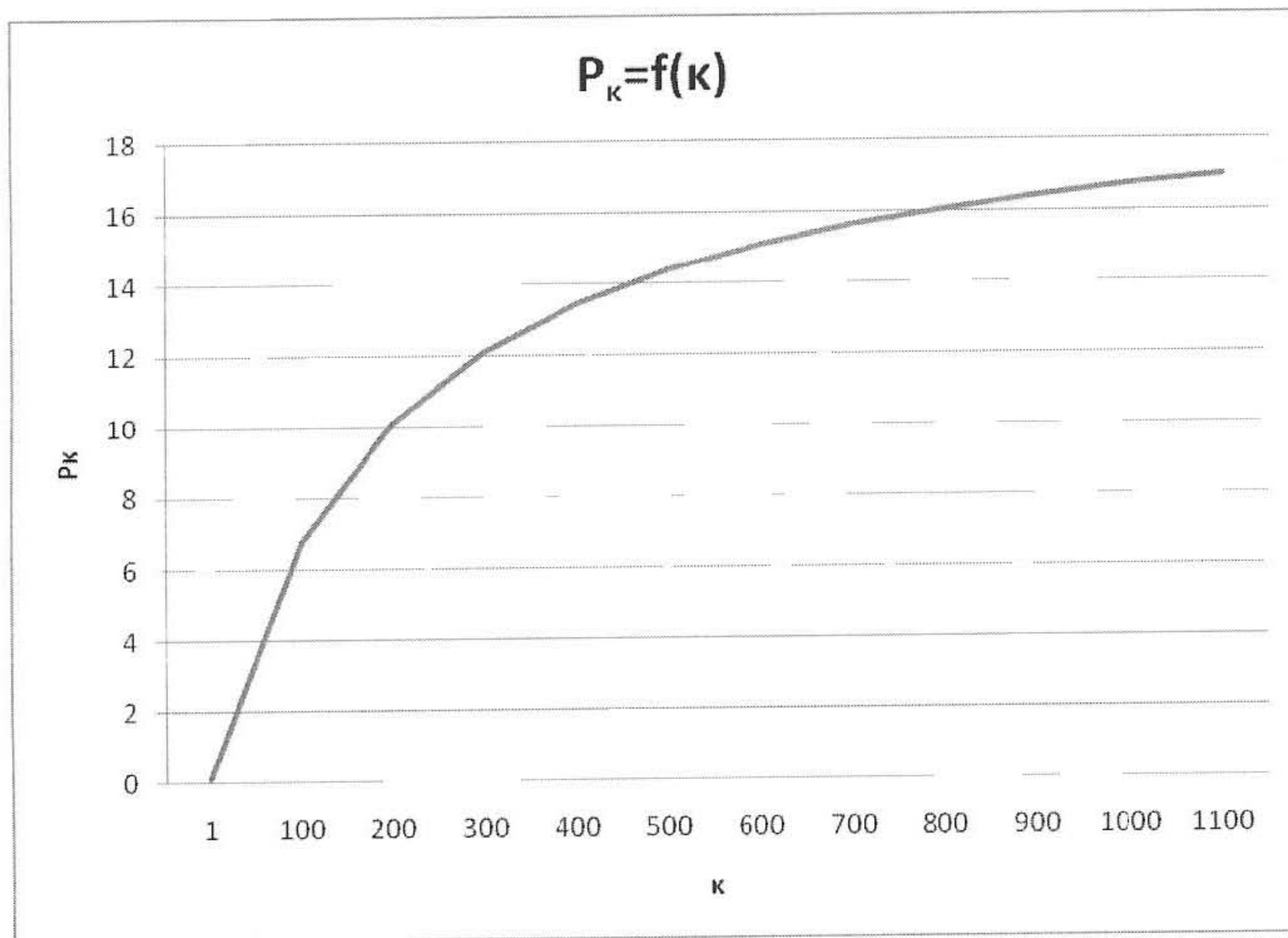
$$\begin{aligned} P_k &= 1/((5 \text{ ms})/(k \times 512 \text{ ψηφιολέξεις}) + 1/(5 \text{ Mψηφιολέξεις το δευτερόλεπτο})) = \\ &= 1/(5/(k \times 512) + 1/5000) \text{ ψηφιολέξεις ανά ms} = \\ &= [1/(5/(k \times 512) + 1/5000)]/1000 \text{ Mψηφιολέξεις ανά δευτερόλεπτο} \end{aligned}$$



Σχήμα 6.8.1 Ρυθμός μεταφοράς δεδομένων εισόδου/εξόδου σε Μψηφιολέξεις το δευτερόλεπτο για μέγιστο ρυθμό μεταφοράς 5 Μψηφιολέξεις το δευτερόλεπτο.

β. Ο μέγιστος ρυθμός μεταφοράς είναι 20 Μψηφιολέξεις το δευτερόλεπτο

$$\begin{aligned}
 P_k &= 1/((5 \text{ ms})/(\kappa \times 512 \text{ ψηφιολέξεις}) + 1/(20 \text{ Μψηφιολέξεις το δευτερόλεπτο})) = \\
 &= 1/(5/(\kappa \times 512) + 1/20.000) \text{ ψηφιολέξεις ανά ms} = \\
 &= [1/(5/(\kappa \times 512) + 1/20.000)]/1000 \text{ Μψηφιολέξεις ανά δευτερόλεπτο.}
 \end{aligned}$$



Σχήμα 6.8.2 Ρυθμός μεταφοράς δεδομένων εισόδου/εξόδου σε Μψηφιολέξεις το δευτερόλεπτο για μέγιστο ρυθμό μεταφοράς 20 Μψηφιολέξεις το δευτερόλεπτο.

Συγκρίνοντας τα σχήματα 6.8.1 και 6.8.2 παρατηρούμε ότι όταν ο μέγιστος ρυθμός μεταφοράς είναι μικρότερος τότε ο ρυθμός μεταφοράς δεδομένων εισόδου/εξόδου λαμβάνει τη μέγιστη τιμή του για μικρότερη τιμή του k . Αυτό εξηγείται παρατηρώντας τη σχέση 6.8.1, όταν η τιμή του MPM είναι μικρή τότε ο όρος $1/MPM$ είναι σημαντικός για μικρότερες τιμές του k . Επίσης από τα σχήματα 6.8.1 και 6.8.2 παρατηρούμε ότι μετά από κάποια τιμή του k , η οποία εξαρτάται από την τιμή MPM, δεν έχει νόημα να αυξάνουμε την τιμή του k , δηλαδή την μονάδα της ποσότητας πληροφορίας που διακινούμε από και προς τη μονάδα του δίσκου.

Άσκηση 6.9

Θεωρήστε ένα μαγνητικό δίσκο με τμήματα (sectors) των 512 ψηφιολέξεων, 512 τμήματα ανά ομόκεντρο κύκλο, μέσο χρόνο προσπέλασης 5 ms και ρυθμό ανάγνωσης-εγγραφής 20 Μψηφιολέξεις το δευτερόλεπτο. Αυτή η μονάδα μαγνητικού δίσκου θα χρησιμοποιηθεί σε μία εφαρμογή ειδικού σκοπού στην οποία θέλουμε να ελαχιστοποιήσουμε το χρόνο που χρησιμοποιείται η μονάδα μαγνητικού δίσκου για τη μεταφορά 32 Μψηφιολέξεων χρήσιμων δεδομένων για την εφαρμογή.

Λαμβάνοντας υπόψη τον Πίνακα 6.9.1 να επιλέξετε τη μονάδα πληροφορίας ($2^k \times 512$ ψηφιολέξεις) που πρέπει να διαβάζεται κάθε φορά από τη μονάδα δίσκου. Θεωρήστε ότι ο ρυθμός μεταφοράς της αρτηρίας είναι μεγαλύτερος των 20 Μψηφιολέξεων το δευτερόλεπτο. Αγνοήστε την καθυστέρηση του ελεγκτή του δίσκου.

Πίνακας 6.9.1

κ	μονάδα πληροφορίας που μεταφέρεται	το ποσοστό των δεδομένων που χρησιμοποιείται από τη μονάδα πληροφορίας που μεταφέρεται %
0	512 ψηφιολέξεις	99
1	2×512 ψηφιολέξεις	80
2	4×512 ψηφιολέξεις	58
3	8×512 ψηφιολέξεις	32
4	16×512 ψηφιολέξεις	18
5	32×512 ψηφιολέξεις	10
6	64×512 ψηφιολέξεις	6
7	128×512 ψηφιολέξεις	3,5
8	256×512 ψηφιολέξεις	2
9	512×512 ψηφιολέξεις	1,1

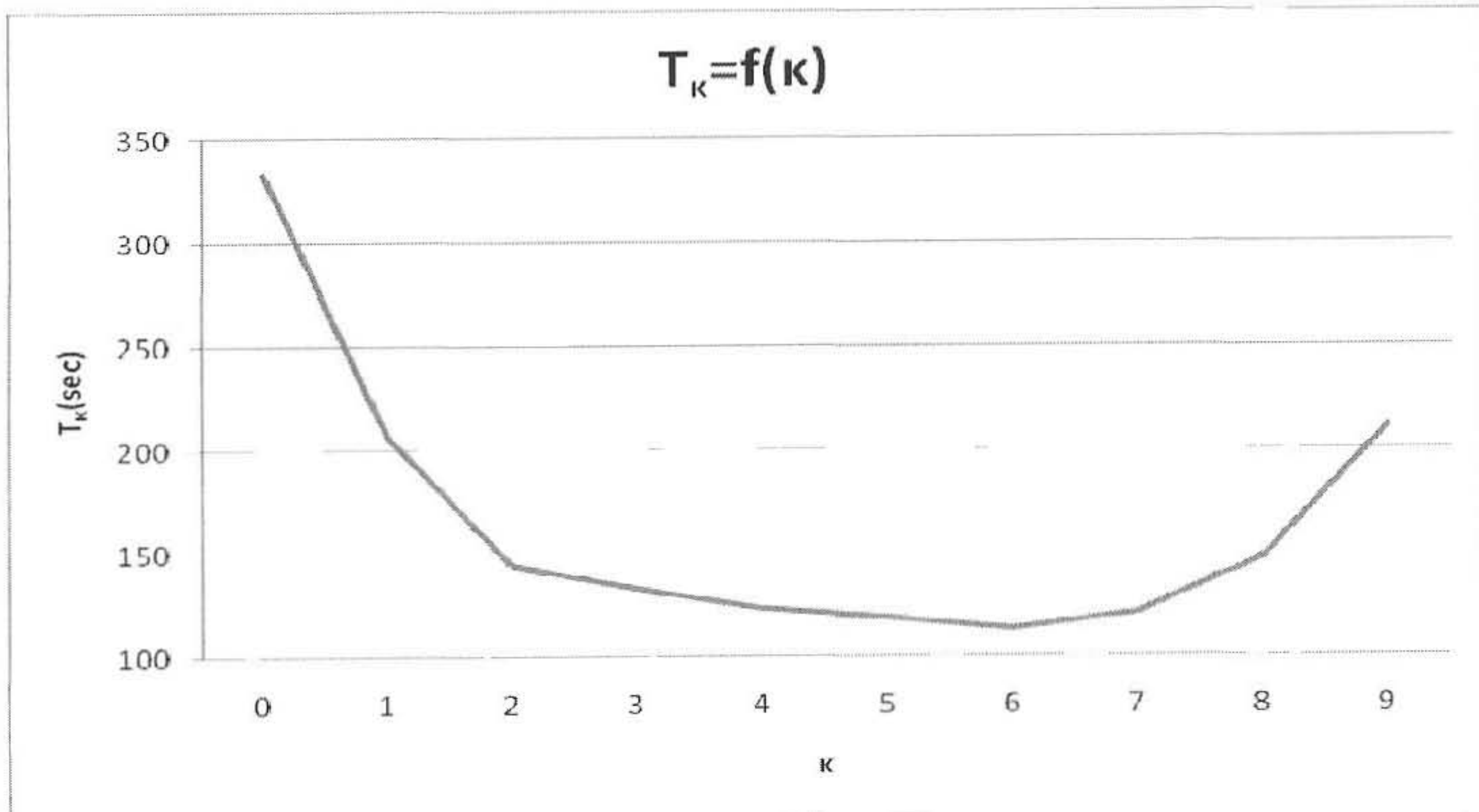
Απάντηση

Θεωρούμε ότι όταν μεταφέρονται ως ομάδα n τμήματα, με $n \leq 512$, τότε τα τμήματα αυτά είναι γειτονικά στον ίδιο ομόκεντρο κύκλο. Από όλα τα γειτονικά δεδομένα που μεταφέρονται, μόνο ένα ποσοστό χρησιμοποιείται. Από τον Πίνακα 6.9.1 βλέπουμε ότι όσο μεγαλύτερη είναι η ποσότητα πληροφορίας που μεταφέρεται, τόσο μικρότερο είναι το ποσοστό των δεδομένων που χρησιμοποιείται. Για να μεταφερθούν 32 Μψηφιολέξεις χρήσιμων δεδομένων απαιτείται να μεταφερθούν συνολικά $(100 \times 32 \text{ Μψηφιολέξεις}) / \text{ΠΧ}$, όπου ΠΧ είναι το ποσοστό των δεδομένων που χρησιμοποιείται από τα δεδομένα που μεταφέρονται. Επομένως ο χρόνος μεταφοράς T_k δίνεται από τη σχέση:

$$\begin{aligned}
 T_k &= (\text{πλήθος μονάδων πληροφορίας που μεταφέρονται}) \times [(\text{χρόνος προσπέλασης}) + \\
 &\quad (2^k \times 512) / (\text{ρυθμός ανάγνωσης})] = \\
 &= \lceil ((100 \times 32 \text{ Μψηφιολέξεις}) / \text{ΠΧ}) / (2^k \times 512 \text{ ψηφιολέξεις}) \rceil \times [(\text{χρόνος προσπέλασης}) + \\
 &\quad (2^k \times 512 \text{ ψηφιολέξεις}) / (\text{ρυθμός ανάγνωσης και εγγραφής})] =
 \end{aligned}$$

$$\begin{aligned}
&= \lceil ((100 \times 32 \times 2^{20} \text{ ψηφιολέξεις}) / \text{ΠΧ}) / (2^k \times 512 \text{ ψηφιολέξεις}) \rceil \times \\
&\quad (5 \text{ ms} + (2^k \times 512 \text{ ψηφιολέξεις}) / (20 \times 10^6 \text{ ψηφιολέξεις/s})) = \\
&= \lceil ((100 \times 32 \times 2^{20}) / \text{ΠΧ}) / (2^k \times 512) \rceil \times (5 \text{ ms} + (2^k \times 512) / (20 \times 10^6) \text{ s}) = \\
&= \lceil ((100 \times 32 \times 2^{20}) / \text{ΠΧ}) / (2^k \times 512) \rceil \times (5 \text{ ms} + 1000 \times (2^k \times 512) / (20 \times 10^6) \text{ ms}) = \\
&= \lceil ((100 \times 32 \times 2^{20}) / \text{ΠΧ}) / (2^k \times 512) \rceil \times (5 + (2^k \times 512) / (20 \times 10^3)) \text{ ms}
\end{aligned}$$

Στο σχήμα 6.9.1 δίνεται ο συνολικός χρόνος μεταφοράς των 32 Μψηφιολέξεων χρήσιμων δεδομένων ως συνάρτηση του k . Παρατηρούμε ότι ο ελάχιστος χρόνος χρησιμοποίησης του δίσκου επιτυγχάνεται για $k=6$, δηλαδή όταν η μονάδα πληροφορίας που μεταφέρεται είναι $2^6 \times 512$ ψηφιολέξεις = $2^6 \times 2^9$ ψηφιολέξεις = 2^{15} ψηφιολέξεις = 32 Κψηφιολέξεις.



Σχήμα 6.9.1

Άσκηση 6.10

Θεωρήστε δύο μαγνητικούς δίσκους Δ1 και Δ2, κάθε ένας με χωρητικότητα 250 Gψηφιολέξεων, 4 επιφάνειες εγγραφής (2 platters), 16.000 κυλίνδρους, τμήματα (sectors) των 512 ψηφιολέξεων, ταχύτητα περιστροφής των δίσκων 5.400 rpm και μέσο χρόνο αναζήτησης 12 ms. Ο Δ1 δεν διαθέτει κρυφή μνήμη δίσκου, ενώ Δ2 διαθέτει κρυφή μνήμη 32 Μψηφιολέξεων, στην οποία αποθηκεύονται ολόκληροι ομόκεντροι κύκλοι του δίσκου. Θεωρήστε ότι ο λόγος επιτυχίας της κρυφής μνήμης είναι $E = 0,9$. Θεωρήστε επίσης ότι όλοι οι ομόκεντροι κύκλοι περιέχουν τον ίδιο αριθμό τμημάτων.

α. Να υπολογίσετε τη βελτίωση του μέσου χρόνου προσπέλασης του Δ2 ως προς τον Δ1.

β. Εάν ο μέγιστος ρυθμός μεταφοράς από τη κρυφή μνήμη του Δ2 (I/O maximum transfer rate) είναι 3 G δυαδικά ψηφία το δευτερόλεπτο και αυτός ο ρυθμός υποστηρίζεται από την αρτηρία εισόδου/εξόδου, και η μονάδα μεταφοράς πληροφορίας από και προς κάθε μία από τις μονάδες δίσκου Δ1 και Δ2 είναι 4 Kψηφιολέξεις, να υπολογίσετε το συνολικό χρόνο ανάγνωσης μίας μονάδας πληροφορίας από τον Δ1 και Δ2 και να τους συγκρίνετε.

Απάντηση

Αφού κάθε μία μονάδα δίσκου έχει χωρητικότητα 250 Gψηφιολέξεις, έχει 4 επιφάνειες εγγραφής (2 platters), 16.000 κυλίνδρους και τμήματα (sectors) των 512 ψηφιολέξεων, συνεπάγεται ότι κάθε επιφάνεια έχει χωρητικότητα $(250 \times 2^{30} \text{ψηφιολέξεις})/4 = 250 \times 2^{28} \text{ψηφιολέξεις}$. Κάθε ομόκεντρος κύκλος θα έχει $(250 \times 2^{28} \text{ψηφιολέξεις})/16.000 = 4.194.304 \text{ψηφιολέξεις}$ και $(4.194.304 \text{ψηφιολέξεις})/(512 \text{ψηφιολέξεις}) = 8.192 \text{τμήματα}$.

α. Στον μαγνητικό δίσκο Δ1 στον οποίο δεν υπάρχει κρυφή μνήμη, ο μέσος χρόνος προσπέλασης $T_{\Delta 1}$ ισούται με το άθροισμα του μέσου χρόνου αναζήτησης T_1 , δηλαδή του μέσου χρόνου που απαιτείται για την τοποθέτηση των κεφαλών στο σωστό κύλινδρο, και του μέσου χρόνου αναμονής T_2 , δηλαδή του μέσου χρόνου που παρέρχεται έως ότου η κεφαλή βρεθεί στην αρχή του τμήματος (sector) που περιέχει τη ζητούμενη πληροφορία. Ο μέσος χρόνος αναζήτησης T_1 μας δίνεται ίσος με 12 ms, δηλαδή $T_1 = 12 \text{ ms}$. Ο μέσος χρόνος αναμονής T_2 , ισούται με το χρόνο που απαιτείται για μισή περιστροφή του δίσκου. Επομένως

$$T_2 = (60 \times 1000)/(5400 \times 2) = 5,55 \text{ ms}.$$

Επομένως ο μέσος χρόνος προσπέλασης του Δ1 είναι $T_{\Delta 1} = T_1 + T_2 = 12 \text{ ms} + 5,55 \text{ ms} = 17,5 \text{ ms}$.

Στο μαγνητικό δίσκο Δ2 θεωρούμε ότι όταν η ζητούμενη πληροφορία δεν υπάρχει στην κρυφή μνήμη του δίσκου, τότε μεταφέρεται το περιεχόμενο ολόκληρου του ομόκεντρου κύκλου που περιέχει τη ζητούμενη πληροφορία από το δίσκο στην κρυφή μνήμη και μετά η ζητούμενη πληροφορία διαβάζεται από την κρυφή μνήμη. Έστω ότι $T_{\text{κρυφή}}$ είναι ο χρόνος προσπέλασης της κρυφής μνήμης και $T_{\text{ομ}}$ είναι ο χρόνος προσπέλασης ενός ομόκεντρου κύκλου και μεταφοράς της πληροφορίας του από το δίσκο στην κρυφή μνήμη. Τότε ο μέσος χρόνος προσπέλασης του δίσκου Δ2 είναι:

$$T_{\Delta 2} = E \times T_{\text{κρυφή}} + (1-E) \times (T_{\text{ομ}} + T_{\text{κρυφή}})$$

Επειδή ο χρόνος προσπέλασης της κρυφής μνήμης, $T_{\text{κρυφή}}$, είναι τάξεις μεγέθους μικρότερος από αυτόν του χρόνου προσπέλασης του δίσκου, ο όρος $T_{\text{κρυφή}}$ μπορεί να αγνοηθεί στον υπολογισμό του $T_{\Delta 2}$, δηλαδή μπορούμε να θεωρήσουμε $T_{\text{κρυφή}} = 0$.

$$\text{Τότε } T_{\Delta 2} \approx (1-E) \times T_{\text{ομ}}$$

Η τιμή του $T_{\text{ομ}}$ ισούται με το άθροισμα του μέσου χρόνου αναζήτησης T_1 , του μέσου χρόνου T_3 που απαιτείται ώστε η κεφαλή ανάγνωσης/εγγραφής να βρεθεί στην αρχή κάποιου, οποιουδήποτε, τμήματος και του χρόνου μεταφοράς του περιεχομένου ενός ομόκεντρου κύκλου στην κρυφή μνήμη T_4 , δηλαδή $T_{\text{ομ}} = T_1 + T_3 + T_4$

Υπολογίσαμε ότι κάθε ομόκεντρος κύκλος έχει 8.192 τμήματα, επομένως

$$T_3 = (60 \times 1.000) / (2 \times 5.400 \times 8.192) \text{ ms} \approx 0,00068 \text{ ms.}$$

$$T_4 = (60 \times 1.000) / 5.400 \text{ ms} = 11,11 \text{ ms}$$

$$\text{Τότε } T_{\text{ομ}} = T_1 + T_3 + T_4 = 12 \text{ ms} + 0,00068 \text{ ms} + 11,11 \text{ ms} = 23,11068 \text{ ms}$$

Άρα ο μέσος χρόνος προσπέλασης της μονάδας $\Delta 2$ είναι:

$$T_{\Delta 2} \approx (1-E) \times T_{\text{ομ}} = (1 - 0,9) \times 23,11068 \text{ ms} = 0,1 \times 23,11068 \text{ ms} \approx 2,311 \text{ ms.}$$

Συνεπώς η βελτίωση που επιτυγχάνεται με τη χρήση της κρυφής μνήμης είναι $T_{\Delta 1} / T_{\Delta 2} = 17,5 / 2,311 = 7,57$.

β. Οι 4 Κψηφιολέξεις $= 4 \times 2^{10}$ ψηφιολέξεις καταλαμβάνουν $(4 \times 2^{10}) / 512$ τμήματα $= (2^2 \times 2^{10}) / 2^9$ τμήματα $= 8$ τμήματα. Επειδή οι 4 Κψηφιολέξεις αποτελούν τη μονάδα της πληροφορίας που μεταφέρεται από και προς το δίσκο, τα 8 τμήματα θα είναι γειτονικά τμήματα πάνω στον ίδιο ομόκεντρο κύκλο. Επομένως ο συνολικός χρόνος ανάγνωσης είναι:

$$T_{\Delta 1} = (\text{μέσος χρόνος προσπέλασης του δίσκου } \Delta 1) + (\text{χρόνος ανάγνωσης από την επιφάνεια του δίσκου της μονάδας } \Delta 1 \text{ 8 τμημάτων})$$

Στο ερώτημα α υπολογίσαμε ότι ο μέσος χρόνος προσπέλασης του $\Delta 1$ είναι $T_{\Delta 1} = 17,5 \text{ ms}$. Επομένως:

$$T_{\Delta 1} = 17,5 \text{ ms} + (8 \times 60 \times 1.000) / (5.400 \times 8.192) \text{ ms} = 17,5 \text{ ms} + 0,011 \text{ ms} = 17,511 \text{ ms}$$

Ο χρόνος ανάγνωσης από τον $\Delta 2$ 4 Κψηφιολέξεων δίνεται από τη σχέση:

$$\begin{aligned} T_{\Delta 2} &= E \times (\text{χρόνος ανάγνωσης 4 Κψηφιολέξεων από την κρυφή μνήμη του } \Delta 2) + \\ &\quad (1-E) \times [(\text{χρόνος προσπέλασης ενός ομόκεντρου κύκλου και μεταφοράς της πλη-} \\ &\quad \text{ροφορίας του από το δίσκο στην κρυφή μνήμη} + \text{χρόνος ανάγνωσης 4 Κψηφιο-} \\ &\quad \text{λέξεων από την κρυφή μνήμη του } \Delta 2)] = \\ &= (\text{χρόνος ανάγνωσης 4 Κψηφιολέξεων από την κρυφή μνήμη του } \Delta 2) + \\ &\quad (1-E) \times [(\text{χρόνος προσπέλασης ενός ομόκεντρου κύκλου και μεταφοράς της πλη-} \\ &\quad \text{ροφορίας του από το δίσκο στην κρυφή μνήμη}) \end{aligned}$$

Στο ερώτημα α υπολογίσαμε ότι ο χρόνος προσπέλασης ενός ομόκεντρου κύκλου και μεταφοράς της πληροφορίας του από το δίσκο στην κρυφή μνήμη $T_{ομ}$ είναι

$T_{ομ} = 23,11068 \text{ ms}$. Επομένως

$$\begin{aligned} TA_{\Delta 2} &= (8 \times 4 \text{ K δυαδικά ψηφία}) / (3 \text{ G δυαδικά ψηφία}) \text{ s} + (1 - 0,9) \times 23,11068 \text{ ms} = \\ &= (8 \times 4 \times 2^{10} \text{ δυαδικά ψηφία}) / (3 \times 10^9 \text{ δυαδικά ψηφία}) \text{ s} + 0,1 \times 23,11068 \text{ ms} = \\ &= (8 \times 4 \times 2^{10} \text{ δυαδικά ψηφία}) / (3 \times 10^6 \text{ δυαδικά ψηφία}) \text{ ms} + 2,311068 \text{ ms} = \\ &= 0,109 \text{ ms} + 2,311068 \text{ ms} = 2,42 \text{ ms}. \end{aligned}$$

ΚΕΦΑΛΑΙΟ

7

Ασκήσεις για υπερβαθμωτούς επεξεργαστές και επεξεργαστές πολύ μεγάλου μήκους εντολών

Άσκηση 7.1 (7.1 βιβλίου)

Οι μονάδες "Μονάδα Προσωρινής Αποθήκευσης Εντολών, ΜΠΑΕ" και "Μονάδα Αναμονής Αποστολής, ΜΑΑ" είναι σύνολα καταχωρητών τα οποία χρησιμοποιούνται για ενδιάμεση αποθήκευση εντολών που βρίσκονται σε κάποια κατάσταση. Κατά την γνώμη σας ποία είναι η χρησιμότητα αυτών των καταχωρητών;

Απάντηση

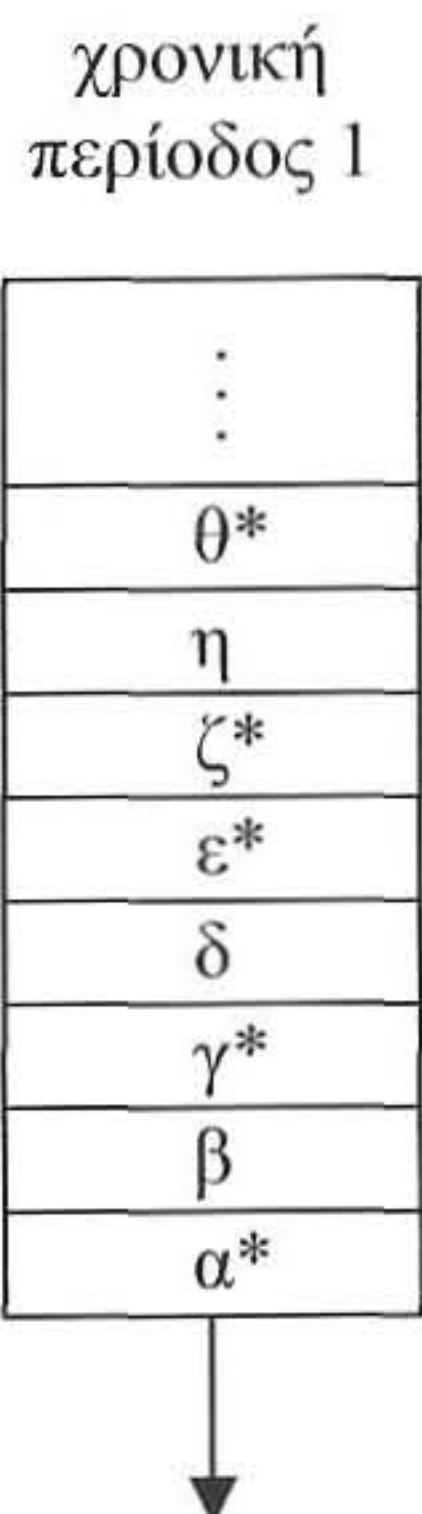
Τα σύνολα αυτών των καταχωρητών οδηγούν στην εξομάλυνση του ρυθμού μεταφοράς εντολών μεταξύ των μονάδων, με αποτέλεσμα να γίνεται καλύτερη εκμετάλλευση των λειτουργικών μονάδων. Για παράδειγμα ας αναφερθούμε στο σχήμα 7.3 του βιβλίου. Σε κάποιο κύκλο είναι δυνατόν οι εντολές που πρέπει να προσκομιστούν να μην υπάρχουν στην κρυφή μνήμη. Εάν δεν υπήρχε η Μονάδα Προσωρινής Αποθήκευσης Εντολών, ΜΠΑΕ, η Μονάδα Αποκωδικοποίησης, Ελέγχου Εξαρτήσεων και Αποστολής Εντολών, ΜΑΕΕΑ, δε θα μπορούσε να στείλει καμία εντολή προς εκτέλεση στις Λειτουργικές Μονάδες. Τώρα όμως αυξάνεται η πιθανότητα να υπάρχουν κάποιες εντολές στην μονάδα ΜΠΑΕ, ώστε η μονάδα ΜΑΕΕΑ να μπορεί να στείλει κάποιες εντολές για εκτέλεση στις Λειτουργικές Μονάδες.

Άσκηση 7.2 (7.2 βιβλίου)

Στο σχήμα 7.2.1 δίνεται το περιεχόμενο της Μονάδας Προσωρινής Αποθήκευσης Εντολών, ΜΠΑΕ. Θεωρήστε παράθυρο τεσσάρων εντολών και ότι το σύμβολο * δηλώνει εντολή που δεν μπορεί να εκτελεστεί λόγω εξαρτήσεων.

- i. Για κάθε μία από τις ακόλουθες τέσσερις περιπτώσεις δηλώστε το περιεχόμενο της ΜΠΑΕ και τις εντολές που θα αποσταλούν για εκτέλεση κατά την τρέχουσα και τις επόμενες τρεις χρονικές μονάδες.
 - α. Θεωρήστε ότι η αποστολή εντολών γίνεται με τη σειρά που εμφανίζονται οι εντολές στο πρόγραμμα και
 - α.1. το παράθυρο εντολών είναι σταθερό.

- α.2. το παράθυρο εντολών είναι μετατοπιζόμενο.
 β. Θεωρήστε ότι η αποστολή εντολών γίνεται εκτός σειράς και
 β.1. το παράθυρο εντολών είναι σταθερό.
 β.2. το παράθυρο εντολών είναι μετατοπιζόμενο.
 ii. Να συγκρίνετε μεταξύ τους τις περιπτώσεις α.1, α.2, β.1 και β.2.



Σχήμα 7.2.1 Τρέχουσα κατάσταση της ΜΠΑΕ.

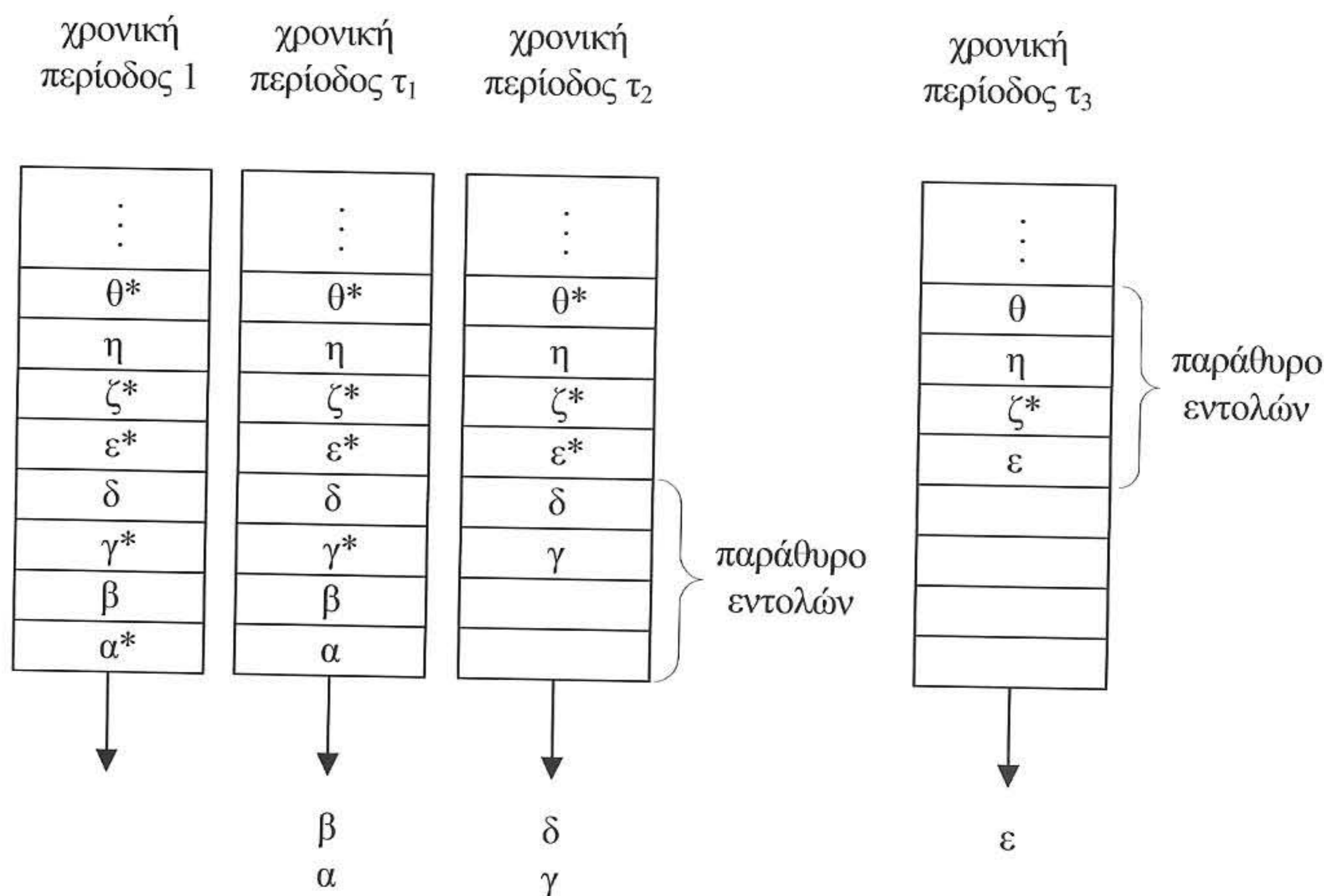
Απάντηση

Στα σχήματα που ακολουθούν τ_i είναι η πρώτη χρονική περίοδος κατά την οποία η εντολή που βρίσκεται πιο κοντά στην έξοδο της ΜΠΑΕ μπορεί να αποσταλεί προς εκτέλεση.

i.

α.1.

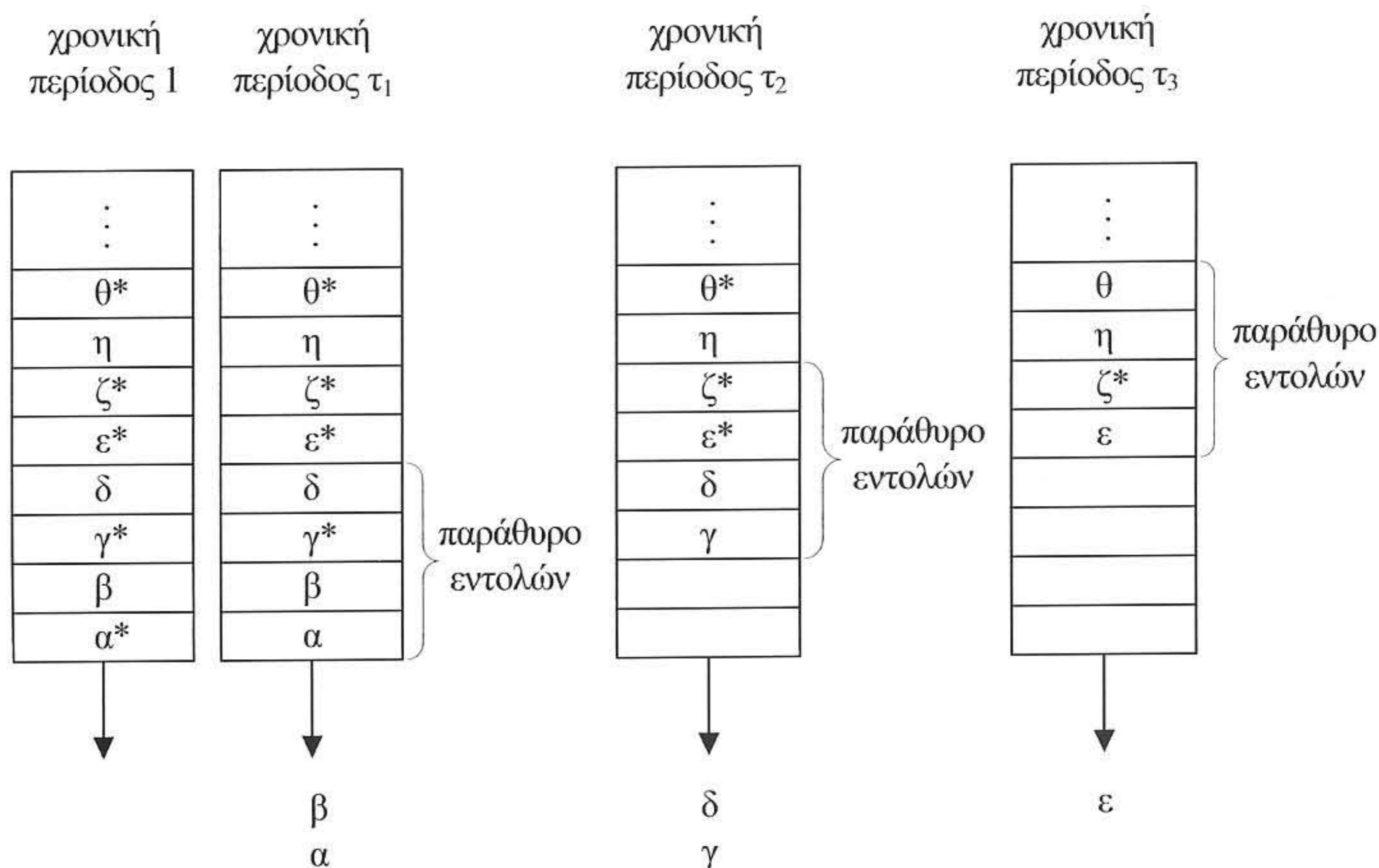
Επειδή η αποστολή εντολών γίνεται με τη σειρά που εμφανίζονται οι εντολές στο πρόγραμμα και την πρώτη χρονική περίοδο η πρώτη εντολή, λόγω εξαρτήσεων, δεν μπορεί να εκτελεστεί, δεν αποστέλλεται καμία εντολή προς εκτέλεση (βλέπε σχήμα 7.2.2). Τη χρονική περίοδο τ_1 οι δύο πρώτες εντολές μπορούν να εκτελεστούν και στέλνονται προς εκτέλεση. Τη χρονική περίοδο τ_2 οι δύο πρώτες εντολές μπορούν να εκτελεστούν και στέλνονται προς εκτέλεση. Επομένως όλες οι εντολές του τρέχοντος παράθυρου έχουν αποσταλεί προς εκτέλεση, οπότε την επόμενη χρονική περίοδο το παράθυρο μετατοπίζεται κατά τέσσερις θέσεις. Τη χρονική περίοδο τ_3 η εντολή ε στέλνεται προς εκτέλεση.



Σχήμα 7.2.2 Η αποστολή εντολών γίνεται με τη σειρά που εμφανίζονται οι εντολές στο πρόγραμμα και το παράθυρο εντολών είναι σταθερό. τ_1 είναι η πρώτη χρονική περίοδος κατά την οποία η εντολή που βρίσκεται πιο κοντά στην έξοδο της ΜΠΑΕ μπορεί να αποσταλεί προς εκτέλεση. Θεωρήσαμε ότι τη χρονική περίοδο τ_3 έπαψε να είναι εξαρτημένη όχι μόνο η εντολή ϵ , αλλά και η εντολή θ .

α.2.

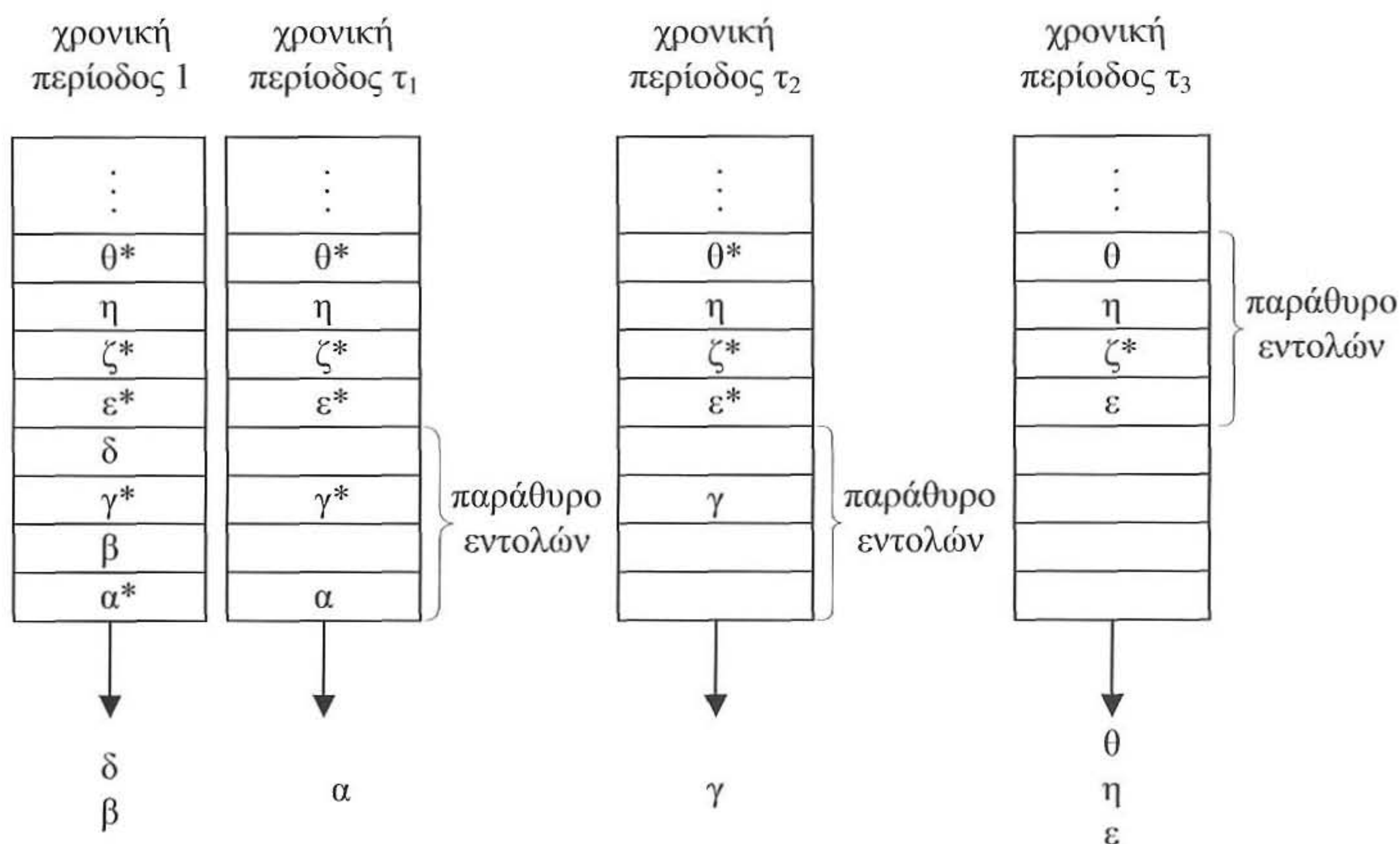
Επειδή η αποστολή εντολών γίνεται με τη σειρά που εμφανίζονται οι εντολές στο πρόγραμμα και την πρώτη χρονική περίοδο η πρώτη εντολή, λόγω εξαρτήσεων, δεν μπορεί να εκτελεστεί, δεν αποστέλλεται καμία εντολή προς εκτέλεση (βλέπε σχήμα 7.2.3). Τη χρονική περίοδο τ_1 οι δύο πρώτες εντολές μπορούν να εκτελεστούν, επομένως στέλνονται προς εκτέλεση και το παράθυρο μετατοπίζεται κατά δύο θέσεις. Τη χρονική περίοδο τ_2 οι δύο πρώτες εντολές μπορούν να εκτελεστούν και στέλνονται προς εκτέλεση. Το παράθυρο μετατοπίζεται και πάλι κατά δύο θέσεις. Τη χρονική περίοδο τ_3 η εντολή ϵ στέλνεται προς εκτέλεση.



Σχήμα 7.2.3 Η αποστολή εντολών γίνεται με τη σειρά που εμφανίζονται οι εντολές στο πρόγραμμα και το παράθυρο εντολών είναι μετατοπιζόμενο. τ_1 είναι η πρώτη χρονική περίοδος κατά την οποία η εντολή που βρίσκεται πιο κοντά στην έξοδο της ΜΠΑΕ μπορεί να αποσταλεί προς εκτέλεση. Θεωρήσαμε ότι τη χρονική περίοδο τ_3 έπαψε να είναι εξαρτημένη όχι μόνο η εντολή ϵ , αλλά και η εντολή θ .

β.1.

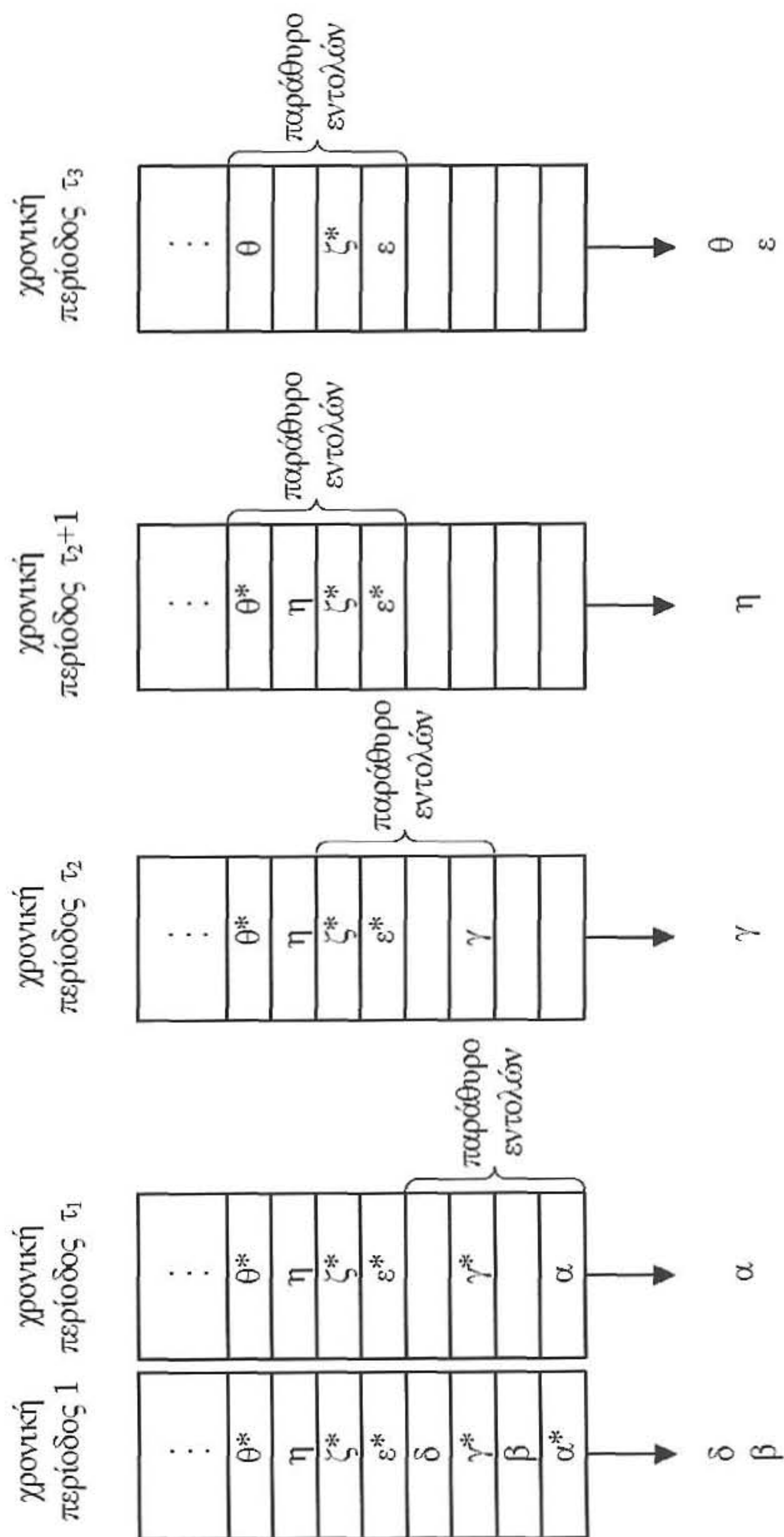
Επειδή η αποστολή εντολών γίνεται εκτός σειράς, οι εντολές β και δ , οι οποίες βρίσκονται στο παράθυρο και μπορούν να σταλούν προς εκτέλεση, στέλνονται προς εκτέλεση (βλέπε σχήμα 7.2.4). Τη χρονική περίοδο τ_1 η πρώτη εντολή α μπορεί να εκτελεστεί και στέλνεται προς εκτέλεση. Τη χρονική περίοδο τ_2 η εντολή γ μπορεί να εκτελεστεί και στέλνεται προς εκτέλεση. Επομένως όλες οι εντολές του τρέχοντος παραθύρου έχουν αποσταλεί προς εκτέλεση, οπότε την επόμενη χρονική περίοδο το παράθυρο μετατοπίζεται κατά τέσσερις θέσεις. Τη χρονική περίοδο τ_3 οι εντολές ϵ , η και θ στέλνονται προς εκτέλεση.



Σχήμα 7.2.4 Η αποστολή εντολών γίνεται εκτός σειράς αλλά το παράθυρο εντολών είναι σταθερό. τ_i είναι η πρώτη χρονική περίοδος κατά την οποία η εντολή που βρίσκεται πιο κοντά στην έξοδο της ΜΠΑΕ μπορεί να αποσταλεί προς εκτέλεση. Θεωρήσαμε ότι τη χρονική περίοδο τ_3 έπαψε να είναι εξαρτημένη όχι μόνο η εντολή ε , αλλά και η εντολή θ .

β.2.

Επειδή η αποστολή εντολών γίνεται εκτός σειράς, οι εντολές β και δ οι οποίες βρίσκονται στο παράθυρο και μπορούν να σταλούν προς εκτέλεση, στέλνονται προς εκτέλεση (βλέπε σχήμα 7.2.5). Τη χρονική περίοδο τ_1 η πρώτη εντολή α μπορεί να εκτελεστεί, επομένως στέλνεται προς εκτέλεση και το παράθυρο μετατοπίζεται κατά δύο θέσεις. Τη χρονική περίοδο τ_2 η εντολή γ μπορεί να εκτελεστεί, επομένως στέλνεται προς εκτέλεση και το παράθυρο μετατοπίζεται κατά δύο θέσεις. Τη χρονική περίοδο τ_2+1 η εντολή η μπορεί να εκτελεστεί, επομένως στέλνεται προς εκτέλεση. Τη χρονική περίοδο τ_3 οι εντολές ε και θ στέλνονται προς εκτέλεση.



Σχήμα 7.2.5 Η αποστολή εντολών γίνεται εκτός σειράς και το παράθυρο εντολών είναι μετατοπιζόμενο. τ_i είναι η πρώτη χρονική περίοδος κατά την οποία η εντολή που βρίσκεται πιο κοντά στην έξοδο της ΜΠΑΕ μπορεί να αποσταλεί προς εκτέλεση. Θεωρήσαμε ότι τη χρονική περίοδο τ_3 έπαψε να είναι εξαρτημένη η δ από την εντολή ε , αλλά και η εντολή θ .

ii. Συνήθως όταν το παράθυρο εντολών είναι μετατοπιζόμενο κάποιες εντολές αποστέλλονται για εκτέλεση νωρίτερα από ότι στην περίπτωση που το παράθυρο είναι σταθερό, κοιτάξτε για παράδειγμα τα σχήματα 7.2.4 και 7.2.5. Ωστόσο στην περίπτωση των σχημάτων 7.2.2 και 7.2.3 η κατανομή των εξαρτημένων και των μη εξαρτημένων εντολών είναι τέτοια που σε κάθε χρονική περίοδο αποστέλλονται για εκτέλεση οι ίδιες ακριβώς εντολές. Επίσης, όταν η αποστολή εντολών προς εκτέλεση γίνεται εκτός σειράς, συνήθως κάποιες εντολές αποστέλλονται για εκτέλεση νωρίτερα από ότι στην περίπτωση στην οποία η αποστολή των εντολών προς εκτέλεση γίνεται με την σειρά που εμφανίζονται στο πρόγραμμα, κοιτάξτε για παράδειγμα τα σχήματα 7.2.2 και 7.2.4.

Άσκηση 7.3 (7.3 βιβλίου)

Θεωρήστε ότι το τμήμα προγράμματος που ακολουθεί βρίσκεται στη Μονάδα Προσωρινής Αποθήκευσης Εντολών ενός υπερβαθμωτού (superscalar) επεξεργαστή με ρυθμό αποστολής εντολών 4, ο οποίος διαθέτει μεταξύ άλλων μία μονάδα Προσπέλασης Δεδομένων/Αποθήκευσης Αποτελεσμάτων (ΠΔ/ΑΑ) με ικανότητα χειρισμού μιας απαίτησης ανά κύκλο ρολογιού, μία Αριθμητική Λογική Μονάδα (ΑΛΜ) με καθυστέρηση ενός κύκλου ρολογιού και μία μονάδα πολλαπλασιασμού με καθυστέρηση 4 κύκλων ρολογιού. Για κάθε μία από τις ακόλουθες περιπτώσεις να συμπληρώσετε ένα πίνακα (όπως ο Πίνακας 7.3.1) ώστε να φαίνεται σε κάθε κύκλο ρολογιού ποιες εντολές αποστέλλονται για εκτέλεση και σε ποια λειτουργική μονάδα.

α. Η αποστολή των εντολών στις λειτουργικές μονάδες προς εκτέλεση γίνεται με προκαθορισμένη σειρά (in-order) και έχουμε μετατοπιζόμενο παράθυρο εντολών.

β. Η αποστολή των εντολών στις λειτουργικές μονάδες προς εκτέλεση γίνεται εκτός σειράς (out-of-order) και έχουμε μετατοπιζόμενο παράθυρο εντολών.

Τμήμα προγράμματος

Load r1, a0	
Load r2, b0	
Mul r3, r1, r2	$r3 \leftarrow r1 \times r2$
Load r4, c0	
Add r5, r3, r4	$r5 \leftarrow r3 + r4$
Store r5, d0	
Load r6, a1	
Load r7, b1	
Mul r8, r6, r7	

Load r9, c1
 Add r10, r8, r9
 Store r10, d1
 Load r11, a2
 Load r12, b2
 Mul r13, r11, r12
 Load r14, c2
 Add r15, r13, r14
 Store r15, d2

Πίνακας 7.3.1

Κύκλος ρολογιού	ΠΔ/ΑΑ	ΑΛΜ	πολλαπλασιαστής
1			
2			
.	.	.	.
.	.	.	.
.	.	.	.

Απάντηση

Σε κάθε περίπτωση, για κάθε κύκλο ρολογιού δίνουμε τις εντολές οι οποίες είναι υποψήφιος για αποστολή προς εκτέλεση και δηλώνουμε την εξάρτηση μιας εντολής, η οποία θα στέλνονταν προς εκτέλεση εάν δεν υπήρχε η εξάρτηση, με * στην περίπτωση εξάρτησης δεδομένων και με ♦ στην περίπτωση δομικής εξάρτησης. Η εντολή ή οι εντολές οι οποίες αποστέλλονται προς εκτέλεση σ' ένα κύκλο ρολογιού βρίσκονται σε σκιασμένο πλαίσιο.

Για την συμπλήρωση των πινάκων που αντιστοιχούν στην περίπτωση α και β λαμβάνουμε υπόψη μας το πλήθος των μονάδων, τις εξαρτήσεις και το ρυθμό αποστολής εντολών.

α. Η αποστολή των εντολών στις λειτουργικές μονάδες προς εκτέλεση γίνεται με προκαθορισμένη σειρά (in-order) και έχουμε μετατοπιζόμενο παράθυρο εντολών.

κύκλος ρολογιού 1	κύκλος ρολογιού 2	κύκλος ρολογιού 3	κύκλος ρολογιού 4
Load r1, a0	Load r2, b0	Mul r3, r1, r2	Add r5, r3, r4*
Load r2, b0*	Mul r3, r1, r2*	Load r4, c0	Store r5, d0
Mul r3, r1, r2	Load r4, c0	Add r5, r3, r4*	Load r6, a1
Load r4, c0	Add r5, r3, r4	Store r5, d0	Load r7, b1

Σημειώστε ότι στον πρώτο κύκλο υπάρχει εξάρτηση δεδομένων της εντολής «Mul r3, r1, r2» από την εντολή «Load r2, b0». Ωστόσο δεν τη σημειώνουμε διότι, αφού η απόστολή των εντολών στις λειτουργικές μονάδες προς εκτέλεση γίνεται με προκαθορισμένη σειρά και η εντολή «Load r2, b0» δεν μπορεί να σταλεί προς εκτέλεση, ακόμη και εάν δεν υπήρχε η εξάρτηση δεδομένων η εντολή «Mul r3, r1, r2» δεν θα μπορούσε να απόσταλεί προς εκτέλεση.

κύκλος ρολογιού 5	κύκλος ρολογιού 6	κύκλος ρολογιού 7	κύκλος ρολογιού 8
Add r5, r3, r4*	Add r5, r3, r4*	Add r5, r3, r4	Store r5, d0
Store r5, d0	Store r5, d0	Store r5, d0*	Load r6, a1*
Load r6, a1	Load r6, a1	Load r6, a1	Load r7, b1
Load r7, b1	Load r7, b1	Load r7, b1	Mul r8, r6, r7

κύκλος ρολογιού 9	κύκλος ρολογιού 10	κύκλος ρολογιού 11	κύκλος ρολογιού 12
Load r6, a1	Load r7, b1	Mul r8, r6, r7	Add r10, r8, r9*
Load r7, b1*	Mul r8, r6, r7*	Load r9, c1	Store r10, d1
Mul r8, r6, r7	Load r9, c1	Add r10, r8, r9*	Load r11, a2
Load r9, c1	Add r10, r8, r9	Store r10, d1	Load r12, b2

κύκλος ρολογιού 13	κύκλος ρολογιού 14	κύκλος ρολογιού 15	κύκλος ρολογιού 16
Add r10, r8, r9*	Add r10, r8, r9*	Add r10, r8, r9	Store r10, d1
Store r10, d1	Store r10, d1	Store r10, d1*	Load r11, a2 [♦]
Load r11, a2	Load r11, a2	Load r11, a2	Load r12, b2
Load r12, b2	Load r12, b2	Load r12, b2	Mul r13, r11, r12

κύκλος ρολογιού 17	κύκλος ρολογιού 18	κύκλος ρολογιού 19	κύκλος ρολογιού 20
Load r11, a2	Load r12, b2	Mul r13, r11, r12	Add r15, r13, r14*
Load r12, b2 [♦]	Mul r13, r11, r12*	Load r14, c2	Store r15, d2
Mul r13, r11, r12	Load r14, c2	Add r15, r13, r14*	
Load r14, c2	Add r15, r13, r14	Store r15, d2	

κύκλος ρολογιού 21	κύκλος ρολογιού 22	Κύκλος ρολογιού 23	κύκλος ρολογιού 24
Add r15, r13, r14*	Add r15, r13, r14*	Add r15, r13, r14	Store r15, d2
Store r15, d2	Store r15, d2	Store r15, d2*	

Από τους παραπάνω πίνακες είναι εύκολο να συμπληρώσουμε τον επόμενο πίνακα.

Κύκλος ρολογιού	ΠΔ/ΑΑ	ΑΛΜ	πολλαπλασιαστής
1	Load r1, a0		
2	Load r2, b0		
3	Load r4, c0		Mul r3, r1, r2
4			
5			
6			
7		Add r5, r3, r4	
8	Store r5, d0		
9	Load r6, a1		
10	Load r7, b1		
11	Load r9, c1		Mul r8, r6, r7
12			
13			
14			
15		Add r10, r8, r9	
16	Store r10, d1		
17	Load r11, a2		
18	Load r12, b2		
19	Load r14, c2		Mul r13, r11, r12
20			
21			
22			
23		Add r15, r13, r14	
24	Store r15, d2		

β. Η αποστολή των εντολών στις λειτουργικές μονάδες προς εκτέλεση γίνεται εκτός σειράς (out-of-order) και έχουμε μετατοπιζόμενο παράθυρο εντολών.

κύκλος ρολογιού 1	κύκλος ρολογιού 2	κύκλος ρολογιού 3	κύκλος ρολογιού 4
Load r1, a0	Load r2, b0	Mul r3, r1, r2	Add r5, r3, r4*
Load r2, b0*	Mul r3, r1, r2*	Load r4, c0	Store r5, d0*
Mul r3, r1, r2*	Load r4, c0*	Add r5, r3, r4*	Load r6, a1
Load r4, c0*	Add r5, r3, r4*	Store r5, d0*	Load r7, b1*

κύκλος ρολογιού 5	κύκλος ρολογιού 6	κύκλος ρολογιού 7	κύκλος ρολογιού 8
Add r5, r3, r4*	Add r5, r3, r4*	Add r5, r3, r4	Store r5, d0
Store r5, d0*	Store r5, d0*	Store r5, d0*	
Load r7, b1			Mul r8, r6, r7

κύκλος ρολογιού 9	κύκλος ρολογιού 10	κύκλος ρολογιού 11	κύκλος ρολογιού 12
Load r9, c1	Add r10, r8, r9*	Add r10, r8, r9*	Add r10, r8, r9
Add r10, r8, r9*	Store r10, d1*	Store r10, d1*	Store r10, d1*
Store r10, d1*	Load r11, a2		
Load r11, a2*	Load r12, b2*	Load r12, b2	

κύκλος ρολογιού 13	κύκλος ρολογιού 14	κύκλος ρολογιού 15	κύκλος ρολογιού 16
Store r10, d1	Load r14, c2	Add r15, r13, r14*	Add r15, r13, r14*
	Add r15, r13, r14*	Store r15, d2*	Store r15, d2*
	Store r15, d2*		
Mul r13, r11, r12			

κύκλος ρολογιού 17	κύκλος ρολογιού 18
Add r15, r13, r14	Store r15, d2
Store r15, d2*	

Από τους παραπάνω πίνακες είναι εύκολο να συμπληρώσουμε τον επόμενο πίνακα.

Κύκλος ρολογιού	ΠΔ/ΑΑ	ΑΛΜ	πολλαπλασιαστής
1	Load r1, a0		
2	Load r2, b0		
3	Load r4, c0		Mul r3, r1, r2
4	Load r6, a1		
5	Load r7, b1		
6			
7		Add r5, r3, r4	
8	Store r5, d0		Mul r8, r6, r7
9	Load r9, c1		
10	Load r11, a2		
11	Load r12, b2		
12		Add r10, r8, r9	
13	Store r10, d1		Mul r13, r11, r12
14	Load r14, c2		
15			
16			
17		Add r15, r13, r14	
18	Store r15, d2		

Παρατηρούμε ότι στη δεύτερη περίπτωση η αποστολή των εντολών για εκτέλεση ολοκληρώνεται σε λιγότερους κύκλους ρολογιού.

Άσκηση 7.4

Θεωρήστε την άσκηση 7.3 με μόνη διαφορά ότι η μονάδα Προσπέλασης Δεδομένων/Αποθήκευσης Αποτελεσμάτων (ΠΔ/ΑΑ) έχει την ικανότητα χειρισμού ταυτόχρονα σ' ένα κύκλο ρολογιού μιας απαίτησης ανάγνωσης και μιας απαίτησης εγγραφής.

Απάντηση

Σε κάθε περίπτωση, για κάθε κύκλο ρολογιού δίνουμε τις εντολές οι οποίες είναι υποψήφιες για αποστολή προς εκτέλεση και δηλώνουμε την εξάρτηση μιας εντολής, η οποία θα στέλνονταν προς εκτέλεση εάν δεν υπήρχε η εξάρτηση, με * στην περίπτωση εξάρτησης δεδομένων και με ♦ στην περίπτωση δομικής εξάρτησης. Η εντολή ή οι εντολές οι οποίες αποστέλλονται προς εκτέλεση σ' ένα κύκλο ρολογιού βρίσκονται σε σκιασμένο πλαίσιο.

Για την συμπλήρωση των πινάκων που αντιστοιχούν στην περίπτωση α και β λαμβάνουμε υπόψη μας το πλήθος των μονάδων, τις εξαρτήσεις και το ρυθμό αποστολής εντολών.

α. Η αποστολή των εντολών στις λειτουργικές μονάδες προς εκτέλεση γίνεται με προκαθορισμένη σειρά (in-order) και έχουμε μετατοπιζόμενο παράθυρο εντολών.

κύκλος ρολογιού 1	κύκλος ρολογιού 2	κύκλος ρολογιού 3	κύκλος ρολογιού 4
Load r1, a0	Load r2, b0	Mul r3, r1, r2	Add r5, r3, r4*
Load r2, b0♦	Mul r3, r1, r2*	Load r4, c0	Store r5, d0
Mul r3, r1, r2	Load r4, c0	Add r5, r3, r4*	Load r6, a1
Load r4, c0	Add r5, r3, r4	Store r5, d0	Load r7, b1

κύκλος ρολογιού 5	κύκλος ρολογιού 6	κύκλος ρολογιού 7	κύκλος ρολογιού 8
Add r5, r3, r4*	Add r5, r3, r4*	Add r5, r3, r4	Store r5, d0
Store r5, d0	Store r5, d0	Store r5, d0*	Load r6, a1
Load r6, a1	Load r6, a1	Load r6, a1	Load r7, b1♦
Load r7, b1	Load r7, b1	Load r7, b1	Mul r8, r6, r7

κύκλος ρολογιού 9	κύκλος ρολογιού 10	κύκλος ρολογιού 11	κύκλος ρολογιού 12
Load r7, b1	Mul r8, r6, r7	Add r10, r8, r9*	Add r10, r8, r9*
Mul r8, r6, r7*	Load r9, c1	Store r10, d1	Store r10, d1
Load r9, c1	Add r10, r8, r9*	Load r11, a2	Load r11, a2
Add r10, r8, r9	Store r10, d1	Load r12, b2	Load r12, b2

κύκλος ρολογιού 13	κύκλος ρολογιού 14	κύκλος ρολογιού 15	κύκλος ρολογιού 16
Add r10, r8, r9*	Add r10, r8, r9	Store r10, d1	Load r12, b2
Store r10, d1	Store r10, d1*	Load r11, a2	Mul r13, r11, r12*
Load r11, a2	Load r11, a2	Load r12, b2 [†]	Load r14, c2
Load r12, b2	Load r12, b2	Mul r13, r11, r12	Add r15, r13, r14

κύκλος ρολογιού 17	κύκλος ρολογιού 18	κύκλος ρολογιού 19	κύκλος ρολογιού 20
Mul r13, r11, r12	Add r15, r13, r14*	Add r15, r13, r14*	Add r15, r13, r14*
Load r14, c2	Store r15, d2	Store r15, d2	Store r15, d2
Add r15, r13, r14*			
Store r15, d2			

Κύκλος ρολογιού 21	κύκλος ρολογιού 22
Add r15, r13, r14	Store r15, d2
Store r15, d2*	

Από τους παραπάνω πίνακες είναι εύκολο να συμπληρώσουμε τον επόμενο πίνακα.

Κύκλος ρολογιού	ΠΔ/ΑΑ	ΑΛΜ	πολλαπλασιαστής
1	Load r1, a0		
2	Load r2, b0		
3	Load r4, c0		Mul r3, r1, r2
4			
5			
6			
7		Add r5, r3, r4	
8	Store r5, d0 Load r6, a1		
9	Load r7, b1		
10	Load r9, c1		Mul r8, r6, r7
11			
12			
13			
14		Add r10, r8, r9	
15	Store r10, d1 Load r11, a2		
16	Load r12, b2		
17	Load r14, c2		Mul r13, r11, r12
18			
19			
20			
21		Add r15, r13, r14	
22	Store r15, d2		

Λόγω της ικανότητας της μονάδα Προσπέλασης Δεδομένων/Αποθήκευσης Αποτελεσμάτων (ΠΔ/ΑΑ) χειρισμού ταυτόχρονα μιας απαίτησης ανάγνωσης και μιας απαίτησης εγγραφής σ' ένα κύκλο ρολογιού, το πρόγραμμα εκτελείται σε δύο λιγότερους κύκλους από ότι στην αντίστοιχη περίπτωση της άσκησης 7.3.

β. Η αποστολή των εντολών στις λειτουργικές μονάδες προς εκτέλεση γίνεται εκτός σειράς (out-of-order) και έχουμε μετατοπιζόμενο παράθυρο εντολών.

κύκλος ρολογιού 1	κύκλος ρολογιού 2	κύκλος ρολογιού 3	κύκλος ρολογιού 4
Load r1, a0	Load r2, b0	Mul r3, r1, r2	Add r5, r3, r4*
Load r2, b0 [♦]	Mul r3, r1, r2*	Load r4, c0	Store r5, d0*
Mul r3, r1, r2*	Load r4, c0 [♦]	Add r5, r3, r4*	Load r6, a1
Load r4, c0 [♦]	Add r5, r3, r4*	Store r5, d0*	Load r7, b1 [♦]

κύκλος ρολογιού 5	κύκλος ρολογιού 6	κύκλος ρολογιού 7	κύκλος ρολογιού 8
Add r5, r3, r4*	Add r5, r3, r4*	Add r5, r3, r4	Store r5, d0
Store r5, d0*	Store r5, d0*	Store r5, d0*	
Load r7, b1			Mul r8, r6, r7

κύκλος ρολογιού 9	κύκλος ρολογιού 10	κύκλος ρολογιού 11	κύκλος ρολογιού 12
Load r9, c1	Add r10, r8, r9*	Add r10, r8, r9*	Add r10, r8, r9
Add r10, r8, r9*	Store r10, d1*	Store r10, d1*	Store r10, d1*
Store r10, d1*	Load r11, a2		
Load r11, a2 [♦]	Load r12, b2 [♦]	Load r12, b2	

κύκλος ρολογιού 13	κύκλος ρολογιού 14	κύκλος ρολογιού 15	κύκλος ρολογιού 16
Store r10, d1	Load r14, c2	Add r15, r13, r14*	Add r15, r13, r14*
	Add r15, r13, r14*	Store r15, d2*	Store r15, d2*
	Store r15, d2*		
Mul r13, r11, r12			

κύκλος ρολογιού 17	κύκλος ρολογιού 18
Add r15, r13, r14	Store r15, d2
Store r15, d2*	

Συγκρίνοντας τους παραπάνω πίνακες με τους αντίστοιχους πίνακες της περίπτωσης β της άσκησης 7.3, παρατηρούμε ότι καίτοι το πλήθος των δομικών εξαρτήσεων είναι πολύ μικρότερο, δεν κερδίζουμε σε κύκλους ρολογιού. Η αιτία είναι η ύπαρξη εξαρτήσεων δεδομένων.

Από τους παραπάνω πίνακες είναι εύκολο να συμπληρώσουμε τον επόμενο πίνακα.

Κύκλος ρολογιού	ΠΔ/ΑΑ	ΑΛΜ	πολλαπλασιαστής
1	Load r1, a0		
2	Load r2, b0		
3	Load r4, c0		Mul r3, r1, r2
4	Load r6, a1		
5	Load r7, b1		
6			
7		Add r5, r3, r4	
8	Store r5, d0		Mul r8, r6, r7
9	Load r9, c1		
10	Load r11, a2		
11	Load r12, b2		
12		Add r10, r8, r9	
13	Store r10, d1		Mul r13, r11, r12
14	Load r14, c2		
15			
16			
17		Add r15, r13, r14	
18	Store r15, d2		

Παρατηρούμε ότι στη δεύτερη περίπτωση η αποστολή των εντολών για εκτέλεση ολοκληρώνεται σε λιγότερους κύκλους ρολογιού.

Άσκηση 7.5

Θεωρήστε την άσκηση 7.3 με μόνη διαφορά ότι η μονάδα Προσπέλασης Δεδομένων/Αποθήκευσης Αποτελεσμάτων (ΠΔ/ΑΑ) έχει την ικανότητα χειρισμού ταυτόχρονα σ' ένα κύκλο ρολογιού δύο απαιτήσεων ανάγνωσης ή δύο απαιτήσεων εγγραφής ή μιας απαίτησης ανάγνωσης και μιας απαίτησης εγγραφής.

Απάντηση

Σε κάθε περίπτωση, για κάθε κύκλο ρολογιού δίνουμε τις εντολές οι οποίες είναι υποψήφιες για αποστολή προς εκτέλεση και δηλώνουμε την εξάρτηση, μιας εντολής η οποία θα στέλνονταν προς εκτέλεση εάν δεν υπήρχε η εξάρτηση, με * στην περίπτωση εξάρτησης δεδομένων και με \diamond στην περίπτωση δομικής εξάρτησης. Η εντολή ή οι εντολές οι οποίες αποστέλλονται προς εκτέλεση σ' ένα κύκλο ρολογιού βρίσκονται σε σκιασμένο πλαίσιο.

Για την συμπλήρωση των πινάκων που αντιστοιχούν στην περίπτωση α και β λαμβάνουμε υπόψη μας το πλήθος των μονάδων, τις εξαρτήσεις και το ρυθμό αποστολής εντολών.

α. Η αποστολή των εντολών στις λειτουργικές μονάδες προς εκτέλεση γίνεται με προκαθορισμένη σειρά (in-order) και έχουμε μετατοπιζόμενο παράθυρο εντολών.

κύκλος ρολογιού 1	κύκλος ρολογιού 2	κύκλος ρολογιού 3	κύκλος ρολογιού 4
Load r1, a0	Mul r3, r1, r2	Add r5, r3, r4*	Add r5, r3, r4*
Load r2, b0	Load r4, c0	Store r5, d0	Store r5, d0
Mul r3, r1, r2*	Add r5, r3, r4*	Load r6, a1	Load r6, a1
Load r4, c0	Store r5, d0	Load r7, b1	Load r7, b1

κύκλος ρολογιού 5	κύκλος ρολογιού 6	κύκλος ρολογιού 7	κύκλος ρολογιού 8
Add r5, r3, r4*	Add r5, r3, r4	Store r5, d0	Load r7, b1
Store r5, d0	Store r5, d0*	Load r6, a1	Mul r8, r6, r7*
Load r6, a1	Load r6, a1	Load r7, b1 \diamond	Load r9, c1
Load r7, b1	Load r7, b1	Mul r8, r6, r7	Add r10, r8, r9

κύκλος ρολογιού 9	κύκλος ρολογιού 10	κύκλος ρολογιού 11	κύκλος ρολογιού 12
Mul r8, r6, r7	Add r10, r8, r9*	Add r10, r8, r9*	Add r10, r8, r9*
Load r9, c1	Store r10, d1	Store r10, d1	Store r10, d1
Add r10, r8, r9*	Load r11, a2	Load r11, a2	Load r11, a2
Store r10, d1	Load r12, b2	Load r12, b2	Load r12, b2

κύκλος ρολογιού 13	κύκλος ρολογιού 14	κύκλος ρολογιού 15	κύκλος ρολογιού 16
Add r10, r8, r9	Store r10, d1	Load r12, b2	Mul r13, r11, r12
Store r10, d1*	Load r11, a2	Mul r13, r11, r12*	Load r14, c2
Load r11, a2	Load r12, b2*	Load r14, c2	Add r15, r13, r14*
Load r12, b2	Mul r13, r11, r12	Add r15, r13, r14	Store r15, d2

κύκλος ρολογιού 17	κύκλος ρολογιού 18	κύκλος ρολογιού 19	Κύκλος ρολογιού 20
Add r15, r13, r14*	Add r15, r13, r14*	Add r15, r13, r14*	Add r15, r13, r14
Store r15, d2	Store r15, d2	Store r15, d2	Store r15, d2*

κύκλος ρολογιού 21
Store r15, d2

Από τους παραπάνω πίνακες είναι εύκολο να συμπληρώσουμε τον επόμενο πίνακα.

Κύκλος ρολογιού	ΠΔ/ΑΑ	ΑΛΜ	πολλαπλασιαστής
1	Load r1, a0 Load r2, b0		
2	Load r4, c0		Mul r3, r1, r2
3			
4			
5			
6		Add r5, r3, r4	
7	Store r5, d0 Load r6, a1		
8	Load r7, b1		
9	Load r9, c1		Mul r8, r6, r7
10			
11			
12			
13		Add r10, r8, r9	
14	Store r10, d1 Load r11, a2		
15	Load r12, b2		
16	Load r14, c2		Mul r13, r11, r12
17			
18			
19			
20		Add r15, r13, r14	
21	Store r15, d2		

Λόγω της ικανότητας της μονάδας Προσπέλασης Δεδομένων/Αποθήκευσης Αποτελεσμάτων (ΠΔ/ΑΑ) χειρισμού ταυτόχρονα στον ίδιο κύκλο ρολογιού δύο απαιτήσεων ανάγνωσης ή δύο απαιτήσεων εγγραφής ή μιας απαίτησης ανάγνωσης και μιας απαίτησης εγγραφής, το πρόγραμμα εκτελείται σε τρεις λιγότερους κύκλους από ότι στην αντίστοιχη περίπτωση της άσκησης 7.3 και σε ένα από την αντίστοιχη περίπτωση της άσκησης 7.4.

β. Η αποστολή των εντολών στις λειτουργικές μονάδες προς εκτέλεση γίνεται εκτός σειράς (out-of-order) και έχουμε μετατοπιζόμενο παράθυρο εντολών.

κύκλος ρολογιού 1	κύκλος ρολογιού 2	κύκλος ρολογιού 3
Load r1, a0	Mul r3, r1, r2	Add r5, r3, r4*
Load r2, b0	Load r4, c0	Store r5, d0*
Mul r3, r1, r2*	Add r5, r3, r4*	Load r6, a1
Load r4, c0*	Store r5, d0*	Load r7, b1

κύκλος ρολογιού 4	κύκλος ρολογιού 5	κύκλος ρολογιού 6	κύκλος ρολογιού 7
Add r5, r3, r4*	Add r5, r3, r4*	Add r5, r3, r4	Store r5, d0
Store r5, d0*	Store r5, d0*	Store r5, d0*	
			Mul r8, r6, r7

κύκλος ρολογιού 8	κύκλος ρολογιού 9	κύκλος ρολογιού 10	κύκλος ρολογιού 11
Load r9, c1	Add r10, r8, r9*	Add r10, r8, r9*	Add r10, r8, r9
Add r10, r8, r9*	Store r10, d1*	Store r10, d1*	Store r10, d1*
Store r10, d1*			
Load r11, a2	Load r12, b2		

κύκλος ρολογιού 12	κύκλος ρολογιού 13	κύκλος ρολογιού 14	κύκλος ρολογιού 15
Store r10, d1	Load r14, c2	Add r15, r13, r14*	Add r15, r13, r14*
	Add r15, r13, r14*	Store r15, d2*	Store r15, d2*
	Store r15, d2*		
Mul r13, r11, r12			

κύκλος ρολογιού 16	κύκλος ρολογιού 17
Add r15, r13, r14	Store r15, d2
Store r15, d2*	

Συγκρίνοντας τους παραπάνω πίνακες με τους αντίστοιχους πίνακες της περίπτωσης β της άσκησης 7.3, παρατηρούμε ότι καίτοι το πλήθος των δομικών εξαρτήσεων είναι πολύ μικρότερο, δεν κερδίζουμε σε κύκλους ρολογιού. Η αιτία είναι η ύπαρξη εξαρτήσεων δεδομένων.

Από τους παραπάνω πίνακες είναι εύκολο να συμπληρώσουμε τον επόμενο πίνακα.

Κύκλος ρολογιού	ΠΔ/ΑΑ	ΑΛΜ	πολλαπλασιαστής
1	Load r1, a0 Load r2, b0		
2	Load r4, c0		Mul r3, r1, r2
3	Load r6, a1 Load r7, b1		
4			
5			
6		Add r5, r3, r4	
7	Store r5, d0		Mul r8, r6, r7
8	Load r9, c1 Load r11, a2		
9	Load r12, b2		
10			
11		Add r10, r8, r9	
12	Store r10, d1		Mul r13, r11, r12
13	Load r14, c2		
14			
15			
16		Add r15, r13, r14	
17	Store r15, d2		

Παρατηρούμε ότι στη δεύτερη περίπτωση της άσκησης η αποστολή των εντολών για εκτέλεση ολοκληρώνεται σε λιγότερους κύκλους ρολογιού.

Άσκηση 7.6 (7.4 βιβλίου)

Θεωρήστε ένα επεξεργαστή ΠΜΜΕ (VLIW) με ένα αθροιστή καθυστέρησης μίας χρονικής περιόδου, δύο πολλαπλασιαστές καθυστέρησης δύο χρονικών περιόδων, μία μονάδα προσκόμισης δεδομένων και μία μονάδα εγγραφής αποτελεσμάτων στη μνήμη με καθυστέρηση δύο χρονικών περιόδων κάθε μία. Θεωρήστε επίσης το ακόλουθο πρόγραμμα σε ψευδοκώδικα:

LOAD R4 \leftarrow M[X]

LOAD R9 \leftarrow M[W]

R1 \leftarrow R2 * R6

R1 \leftarrow R4 * R1

R7 \leftarrow R5 + R8

R3 \leftarrow R2 + R5

R10 \leftarrow R3 * R7

R4 \leftarrow R5 + R4

R9 \leftarrow R6 + R4

STORE M[Y] \leftarrow R4

STORE M[Z] \leftarrow R10

Να δώσετε τη μορφή της εντολής στον ανωτέρω επεξεργαστή ΠΜΜΕ. Επίσης να δώσετε ένα πρόγραμμα με το ελάχιστο πλήθος εντολών που μπορεί να παράγει ένας μεταγλωττιστής για τον ανωτέρω επεξεργαστή.

Απάντηση.

Η μορφή της εντολής και το πρόγραμμα που θα παράγει ο μεταγλωττιστής δίνεται στον επόμενο πίνακα.

Πίνακας 7.6.1

Λειτουργικές μονάδες	Αθροιστής			Πολυπλασιαστής 1			Πολυπλασιαστής 2			Μονάδα προσκόμισης δεδομένων	Μονάδα αποθήκευσης δεδομένων	
εντολές	KΠ	KΔ1	KΔ2	KΠ	KΔ1	KΔ2	KΠ	KΔ1	KΔ2	ΚΕ	ΘΑ	ΚΑ
	R7	R5	R8	R1	R2	R6				R4	X	
	R3	R2	R5									
	R4	R5	R4	R1	R4	R1	R10	R3	R7	R9	W	
	R9	R6	R4								Y	R4
											Z	R10

KΔ1, KΔ2: δηλώνουν καταχωρητές που περιέχουν τα δεδομένα που θα χρησιμοποιηθούν στη συγκεκριμένη λειτουργία.

KΠ: δηλώνει τον καταχωρητή στον οποίο θα αποθηκευτεί το αποτέλεσμα της συγκεκριμένης λειτουργίας.

ΚΕ: δηλώνει τον καταχωρητή στον οποίο θα αποθηκευτεί το δεδομένο που προσκομίζεται από τη θέση μνήμης ΘΑ.

ΚΑ: δηλώνει τον καταχωρητή που περιέχει το δεδομένο το οποίο θα αποθηκευτεί στη θέση μνήμης ΘΕ.

Άσκηση 7.7

Θεωρήστε ένα επεξεργαστή ΠΜΜΕ με ένα αθροιστή καθυστέρησης μίας χρονικής περιόδου, δύο πολλαπλασιαστές καθυστέρησης τριών χρονικών περιόδων, μία μονάδα προσκόμισης δεδομένων και μία μονάδα εγγραφής αποτελεσμάτων στην μνήμη με καθυστέρηση δύο χρονικών περιόδων κάθε μία. Θεωρήστε επίσης το ακόλουθο πρόγραμμα σε ψευδοκώδικα.

```
LOAD R4 ← M[X]  
R1 ← R2 * R6  
R7 ← R5 + R8  
R10 ← R3 * R7  
R9 ← R6 + R4  
R3 ← R2 + R5  
STORE M[Y] ← R3
```

Να δώσετε την μορφή της εντολής στον ανωτέρω επεξεργαστή ΠΜΜΕ.

Επίσης να δώσετε το πρόγραμμα με το ελάχιστο πλήθος εντολών που μπορεί να παράγει ένας μεταγλωττιστής για τον ανωτέρω επεξεργαστή.

Απάντηση

Η μορφή της εντολής και το πρόγραμμα που θα παράγει ο μεταγλωττιστής δίνεται στον επόμενο πίνακα.

Πίνακας 7.7.1

Λειτουργ- γικές μονάδες	αθροιστής			Πολλαπλασιαστής 1			Πολλαπλασιαστής 2			Μονάδα προσκόμισης δεδομένων			Μονάδα αποθήκευσης δεδομένων		
εντολές	ΚΠ	ΚΔ1	ΚΔ2	ΚΠ	ΚΔ1	ΚΔ2	ΚΠ	ΚΔ1	ΚΔ2	ΚΕ	ΘΑ	ΘΕ	ΚΑ		
	R7	R5	R8	R1	R2	R6				R4	X				
	R3	R2	R5				R10	R3	R7						
	R9	R6	R4									Y	R3		

ΑΣΚΗΣΕΙΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Το βιβλίο αυτό προσφέρει αναλυτικά λυμένες ασκήσεις που καλύπτουν τα ακόλουθα θέματα:

- Εισαγωγή στη δομή, οργάνωση, λειτουργία και αξιολόγηση των υπολογιστών
- Οργάνωση και διαχείριση της πληροφορίας (δεδομένα, εντολές, RISC, CISC)
- Σχεδίαση της κεντρικής μονάδας επεξεργασίας και της μονάδας ελέγχου
- Σχεδίαση επεξεργαστών μερικώς επικαλυπτόμενων λειτουργιών (pipelining)
- Σύστημα μνήμης (τεχνολογία μνημών, κρυφή, κύρια και βοηθητική μνήμη, ιδεατή μνήμη)
- Αρτηρίες και διαδικασία εισόδου-εξόδου
- Υπερβαθμωτοί (superscalar) επεξεργαστές και επεξεργαστές πολύ μεγάλου μήκους εντολών (VLIW)

Αν και το βιβλίο αυτό γράφτηκε για να υποστηρίξει με ασκήσεις το βιβλίο "Αρχιτεκτονική Υπολογιστών, Δ. Νικολός, Γκιούρδας Εκδοτική, 2008", η πλειονότητα των ασκήσεων μπορεί να χρησιμοποιηθεί σε συνδυασμό με οποιοδήποτε βιβλίο Αρχιτεκτονικής Υπολογιστών.

Ιστοσελίδα υποστήριξης: Πρόσθετες πληροφορίες για τους φοιτητές και βοηθητικό υλικό για τους διδάσκοντες παρέχονται στην ιστοσελίδα του βιβλίου:
http://www.ceid.upatras.gr/tca-lab/nikolos_en/Arch-book.htm

Ο συγγραφέας του βιβλίου **Δημήτρης Νικολός** είναι καθηγητής του Τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών και διευθυντής του Εργαστηρίου Τεχνολογίας και Αρχιτεκτονικής Υπολογιστών. Έχει δημοσιεύσει περισσότερες από 160 ερευνητικές εργασίες σε διεθνή επιστημονικά περιοδικά και συνέδρια, έχει διατελέσει μέλος των επιστημονικών επιτροπών πολλών διεθνών συνεδρίων και έχει συμμετάσχει σε Ελληνικά και Ευρωπαϊκά ερευνητικά και αναπτυξιακά προγράμματα.



Γκιούρδας
Εκδοτική

email: info@giourdas.com
www.giourdas.com

