Αρχιτεκτονική Υπολογιστών ΙΙ

16 -2-2012

Ενδεικτικές απαντήσεις στα θέματα των εξετάσεων

Θέμα 1. Τι γνωρίζετε για την τοπικότητα των αναφορών και ποιών μονάδων του υπολογιστή ή τεχνικών η απόδοση εξαρτάται από αυτή. (Μονάδες: 2)

Απάντηση.

Η δυνατότητα πρόβλεψης των διευθύνσεων μνήμης που θα αναφερθούν στο άμεσο μέλλον, η οποία είναι ουσιώδης για την επιτυχή λειτουργία της ιεραρχικής μνήμης, βασίζεται σε μία κοινή ιδιότητα των προγραμμάτων που καλείται τοπικότητα των αναφορών (locality of references). Σύμφωνα με την τοπικότητα των αναφορών, η πληροφορία (εντολές και δεδομένα) που χρησιμοποιήθηκε πρόσφατα είναι πιθανόν να ξαναχρησιμοποιηθεί στο άμεσο μέλλον και η πληροφορία που βρίσκεται κοντά στην πληροφορία που χρησιμοποιείται τώρα είναι πιθανόν να χρησιμοποιηθεί στο άμεσο μέλλον.

Ένας λόγος της τοπικότητας των αναφορών είναι ότι οι εντολές και σε μικρότερη έκταση τα δεδομένα, γράφονται και στη συνέχεια αποθηκεύονται στη μνήμη του υπολογιστή σχεδόν με τη σειρά με την οποία χρειάζονται κατά τη διάρκεια της εκτέλεσης του προγράμματος. Επίσης στις εφαρμογές η αποθήκευση των δεδομένων γίνεται με τέτοιο τρόπο ώστε να ενισχύεται η τοπικότητα των αναφορών.

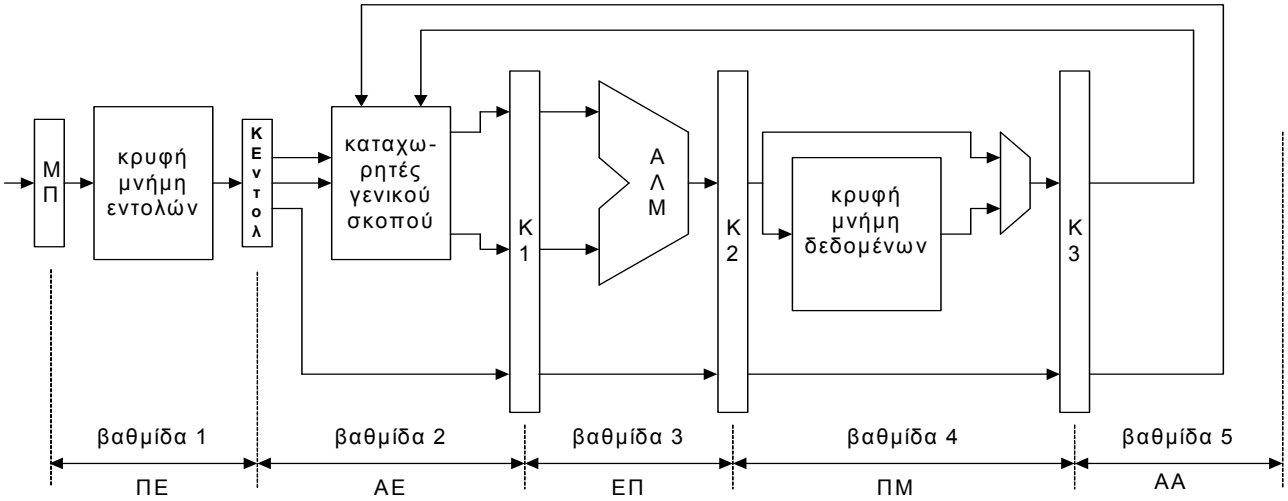
ή

Θέμα 1. Τι γνωρίζετε για την τεχνική του αντιστραμμένου πίνακα σελίδων; (Μονάδες: 2)

Απάντηση.

Ένας τρόπος να μειώσουμε το μέγεθος του πίνακα σελίδων ώστε να διατηρείται ολόκληρος στην κύρια μνήμη και να μην καταλαμβάνει απαγορευτικά μεγάλο τμήμα της, είναι να χρησιμοποιήσουμε την τεχνική του αντιστραμμένου πίνακα σελίδων (inverted page table). Ένας αντιστραμμένος πίνακας σελίδων έχει μία θέση για κάθε πλαίσιο σελίδας της κύριας μνήμης, δηλαδή του χώρου φυσικών διευθύνσεων και όχι μία θέση για κάθε σελίδα του χώρου λογικών διευθύνσεων όπως ο πίνακας σελίδων. Στην περίπτωση αυτή και πάλι ο βασικός καταχωρητής πίνακα σελίδων, ΒΚΠΣ, περιέχει τη διεύθυνση της κύριας μνήμης από την οποία αρχίζει ο αντιστραμμένος πίνακας σελίδων. Μία συνάρτηση διασποράς (hash function) μετασχηματίζει το τμήμα της λογικής διεύθυνσης που δηλώνει τον αριθμό λογικής σελίδας. Το αποτέλεσμα του μετασχηματισμού δηλώνει τη θέση του αντιστραμμένου πίνακα σελίδων που θα προσπελαστεί. Κάθε θέση του αντιστραμμένου πίνακα σελίδων περιέχει πολλές αντιστοιχίσεις αριθμού λογικής σελίδας, ΑΛΣ, σε φυσική διεύθυνση σελίδας, φδΣ, διότι λόγω της συνάρτησης διασποράς που χρησιμοποιούμε, από πολλούς διαφορετικούς αριθμούς λογικών σελίδων μπορεί να προκύψει η ίδια τιμή. Είναι δυνατόν το πλήθος των αριθμών λογικής σελίδας που θα μεταφραστούν στον ίδιο αριθμό να ξεπερνάει τη χωρητικότητα μίας θέσης του αντιστραμμένου πίνακα σελίδων. Αυτό συμβαίνει σπάνια και αντιμετωπίζεται ως απουσία σελίδας. Στην ιδεατή μνήμη του PowerPC, ο οποίος χρησιμοποιεί αντιστραμμένο πίνακα σελίδων, για να μειωθούν οι συνέπειες, έστω και αν αυτή η περίπτωση συμβαίνει σπάνια, χρησιμοποιούν και μία δεύτερη συνάρτηση διασποράς που διαφέρει σημαντικά από την πρώτη. Όταν η ζητούμενη αντιστοίχιση δεν βρίσκεται στην θέση που προέκυψε από την εφαρμογή της πρώτης συνάρτησης διασποράς τότε εφαρμόζεται η δεύτερη συνάρτηση διασποράς για να προκύψει ο αριθμός μίας άλλης θέσης του αντιστραμμένου πίνακα σελίδων στην οποία μπορεί να υπάρχει η αντιστοίχηση. Στην περίπτωση του αντιστραμμένου πίνακα σελίδων κρατάμε πληροφορία μόνο για τις σελίδες που βρίσκονται στην κύρια μνήμη. Το λειτουργικό σύστημα κρατάει πίνακες με την πληροφορία αντιστοίχισης μεταξύ λογικών σελίδων και διευθύνσεων στο μαγνητικό δίσκο.

Θέμα 2. Θεωρήστε το μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών του επόμενου σχήματος.



Εάν το τμήμα προγράμματος που ακολουθεί εκτελείται στον ανωτέρω μηχανισμό μερικώς επικαλυπτόμενων λειτουργιών

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

α. να βρείτε τις εξαρτήσεις που θα προκύψουν. (Μονάδα: 1)

γ. να λύσετε τις εξαρτήσεις χρησιμοποιώντας εντολές NOP. (Μονάδα: 1)

δ. να προτείνετε και να εφαρμόσετε τρόπο μείωσης των απαιτούμενων εντολών NOP σε επίπεδο προγράμματος. (Μονάδα: 1)

Απάντηση.

α. Στο επόμενο σχήμα δίνεται το χρονικό διάγραμμα χρήσης των βαθμίδων κατά την εκτέλεση του προγράμμάτος. Παρατηρούμε ότι η εντολη 3 χρησιμοποιεί το περιεχόμενο του καταχωρητή r2 στον οποίο όμως γράφει η προηγούμενη εντολή, η εντολή 2. Επομένως έχουμε μία εξάρτηση τύπου ΑΜΕ μεταξύ δύο διαδοχικών εντολών η οποία μπορεί να λυθεί παρεμβάλοντας 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 παρεμβάλονται συνολικά τρεις εντολές, επομένως δεν υπάρχει εξάρτηση μεταξύ των εντολών 1 και 3 (βλέπε πρόγραμμα 1). Η εντολή 3 γράφει στον καταχωρητή 3 ενώ η εντολή 4 διαβάζει από τον καταχωρητή 3, επομένως έχουμε εξάρτηση τύπου ΑΜΕ 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 1).

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

β.

Πρόγραμμα 1

```
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
```

γ.

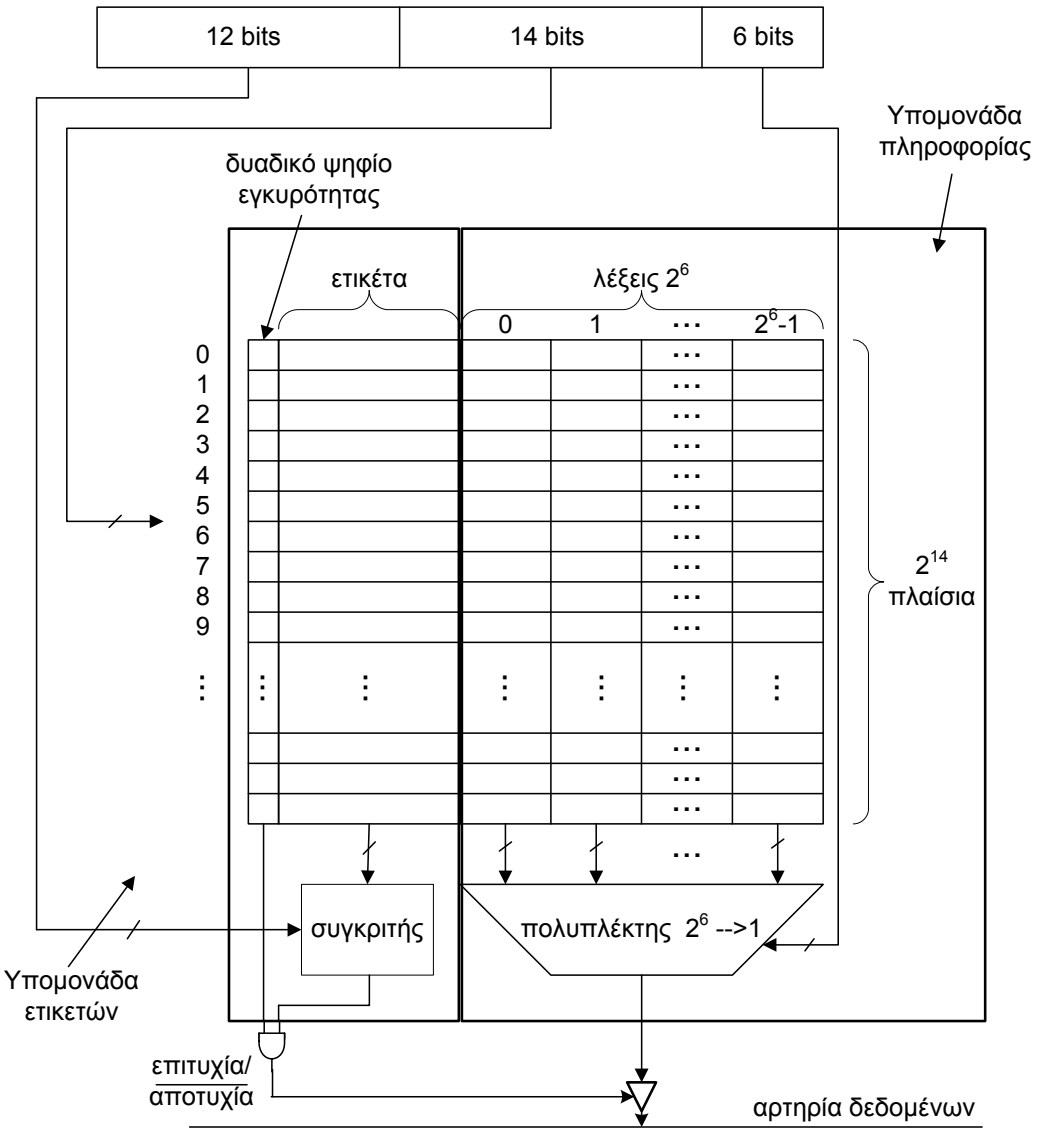
Από την ανωτέρω συζήτηση προκύπτει ότι μεταξύ των δύο εντολών κάθε ενός από τα ζεύγη (2, 3), (3, 4), (6, 7) και (7, 8) του αρχικού προγράμματος πρέπει να παρεμβληθούν δύο ανεξάρτητες εντολές. Επομένως αναδιατάσσοντας τις εντολές με σκοπό τη χρήση του ελάχιστου αριθμού εντολών NOP παίρνουμε το πρόγραμμα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

Πρόγραμμα 2

```
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
```

Θέμα 3. Θεωρήστε σύστημα υπολογιστή με αρτηρία διευθύνσεων των 32 δυαδικών ψηφίων, σύστημα μνήμης με μια ψηφιολέξη ανά θέση μνήμης (δηλαδή μια διεύθυνση αντιστοιχεί σε κάθε ψηφιολέξη) και κρυφή μνήμη με χωρητικότητα 1 Μψηφιολέξεις (1Mbytes) και πλαίσιο των 64 ψηφιολέξεων. Να σχεδιάσετε σε μπλοκ διάγραμμα την κρυφή μνήμη για καθεμία από τις ακόλουθες οργανώσεις και να υπολογίσετε το υλικό (μνήμη, πύλες, πολυπλέκτες κ.λπ.) που απαιτείται για κάθε υλοποίηση. Όλη η απαιτούμενη πληροφορία να φαίνεται στο σχήμα.
α. Οργάνωση μονοσήμαντης απεικόνισης (direct mapped) (Μονάδα: 1)
β. Οργάνωση 2-τρόπων συνόλου συσχέτισης (2-way set associative) (Μονάδα: 1)
Για κάθε μία από τις ανωτέρω οργανώσεις να εξετάσετε ποιές από τις πληροφορίες που βρίσκονται αποθηκευμένες στις διευθύνσεις 00601F4C(16), 00681F4E(16) και 00601F4F(16) μπορούν να βρίσκονται ταυτόχρονα στην κρυφή μνήμη. (Μονάδα: 1)

Απάντηση.
Αφού η χωρητικότητα της κρυφής μνήμης είναι 1 Mbytes = 2^{20} bytes και κάθε πλαίσιο είναι των 64 bytes =2^6 bytes, συνεπάγεται ότι η κρυφή μνήμη έχει $2^{20}/2^6=2^{14}$ πλαίσια. Αφού το σύστημα μνήμης έχει μία ψηφιολέξη (byte) ανά θέση μνήμης και το πλάισιο είναι των $64=2^6$ ψηφιολέξεων, τα 6 λιγότερο σημαντικά δυαδικά ψηφία της διεύθυνσης χρησιμοποιούνται για να δηλώσουν τη διεύθυνση της λέξης μέσα στο πλαίσιο.
α. Οργάνωση μονοσήμαντης απεικόνισης (direct mapped)
Αφού η κρυφή μνήμη έχει 2^{14} πλαίσια, τα επόμενα 14 δυαδικά ψηφία χρησιμοποιούνται για να δηλώσουν τη διεύθυνση του πλασίου. Τα υπόλοιπα 32-14-6=12 δυαδικά ψηφία αποτελούν την ετικέτα.



Για την υλοποίηση αυτής της κρυφής μνήμης απαιτείται μνήμη χωρητικότητας (2^{14} πλαίσια) × (64 δυαδικά ψηφία ανά πλαίσιο) για την αποθήκευση των δεδομένων + (2^{14} πλαίσια) × (12 δυαδικά ψηφία ανά πλάισιο) για την αποθήκευση των ετικετών + (2^{14} πλαίσια) × (1 δυαδικό ψηφίο ανά πλαίσιο) για την αποθήκευση του δυαδικού ψηφίου εγκυρότητας. Επίσης απαιτείται ένας συγκριτής των 12 δυαδικών ψηφίων, μία πύλη AND, 8 πολυπλέκτες 64 → 1 και 8 στοιχεία τριών καταστάσεων.

00601F4C(16): διεύθυνση α
0000 0000 0110 0000 0001 1111 0100 1100

00681F4E(16): διεύθυνση β
0000 0000 0110 1000 0001 1111 0100 1110

00601F4F(16): διεύθυνση γ
0000 0000 0110 0000 0001 1111 0100 1111

Οι διευθύνσεις α και γ ανήκουν στο ίδιο μπλοκ της κύριας μνήμης (έχουν την ίδια διεύθυνση πλαισίου και την ίδια ετικέτα), άρα όταν μεταφέρεται το περιεχόμενο του μπλοκ στην κρυφή μνήμη θα βρίσκονται σ' αυτή τα περιεχόμενα και των δύο διευθύνσεων. Η διεύθυνση β έχει διαφορετική διεύθυνση πλαισίου από τις προηγούμενες, άρα και αυτής τα περιεχόμενο μπορεί να βρίσκεται ταυτόχρονα στην κρυφή μνήμη με τα περιεχόμενα των άλλων δύο.

β. Οργάνωση 2-τρόπων συνόλου συσχέτισης (2-way set associative)

Αφού η κρυφή μνήμη έχει 2^{14} πλαίσια και 2 πλαίσια ανά σύνολο συνεπάγεται ότι έχει 2^{13} σύνολα. Επομένως μετά τα 6 λιγότερο σημαντικά δυαδικά ψηφία της διεύθυνσης τα οποία δηλώνουν τη διεύθυνση της λέξης μέσα στο πλαίσιο, τα 13 δυαδικά ψηφία θα δηλώνουν τη διεύθυνση συνόλου και τα υπόλοιπα 32-13-6=13 δυαδικά ψηφία θα αποτελούν την ετικέτα. Στο σχήμα δίνεται πιο κάτω.

Για την υλοποίηση αυτής της κρυφής μνήμης απαιτείται μνήμη χωρητικότητας 2× (2^{13} πλαίσια) × (64 δυαδικά ψηφία ανά πλαίσιο) για την αποθήκευση των δεδομένων + 2× (2^{13} πλαίσια) × (13 δυαδικά ψηφία ανά πλαίσιο) για την αποθήκευση των ετικετών + 2× (2^{13} πλαίσια) × (1 δυαδικό ψηφίο ανά πλάισιο) για την αποθήκευση του δυαδικού ψηφίου εγκυρότητας + το υλικό που απαιτείται για τη υλοποίηση της τεχνικής αντικατάστασης πλαισίων της κρυφής μνήμης (το υλικό αυτό δεν φαίνεται στο σχήμα). Επίσης απαιτούνται δύο συγκριτές των 13 δυαδικών ψηφίων ο κάθε ένας, δύο πύλες AND, 2×8 πολυπλέκτες 64 → 1 και 2×8 στοιχεία τριών καταστάσεων.

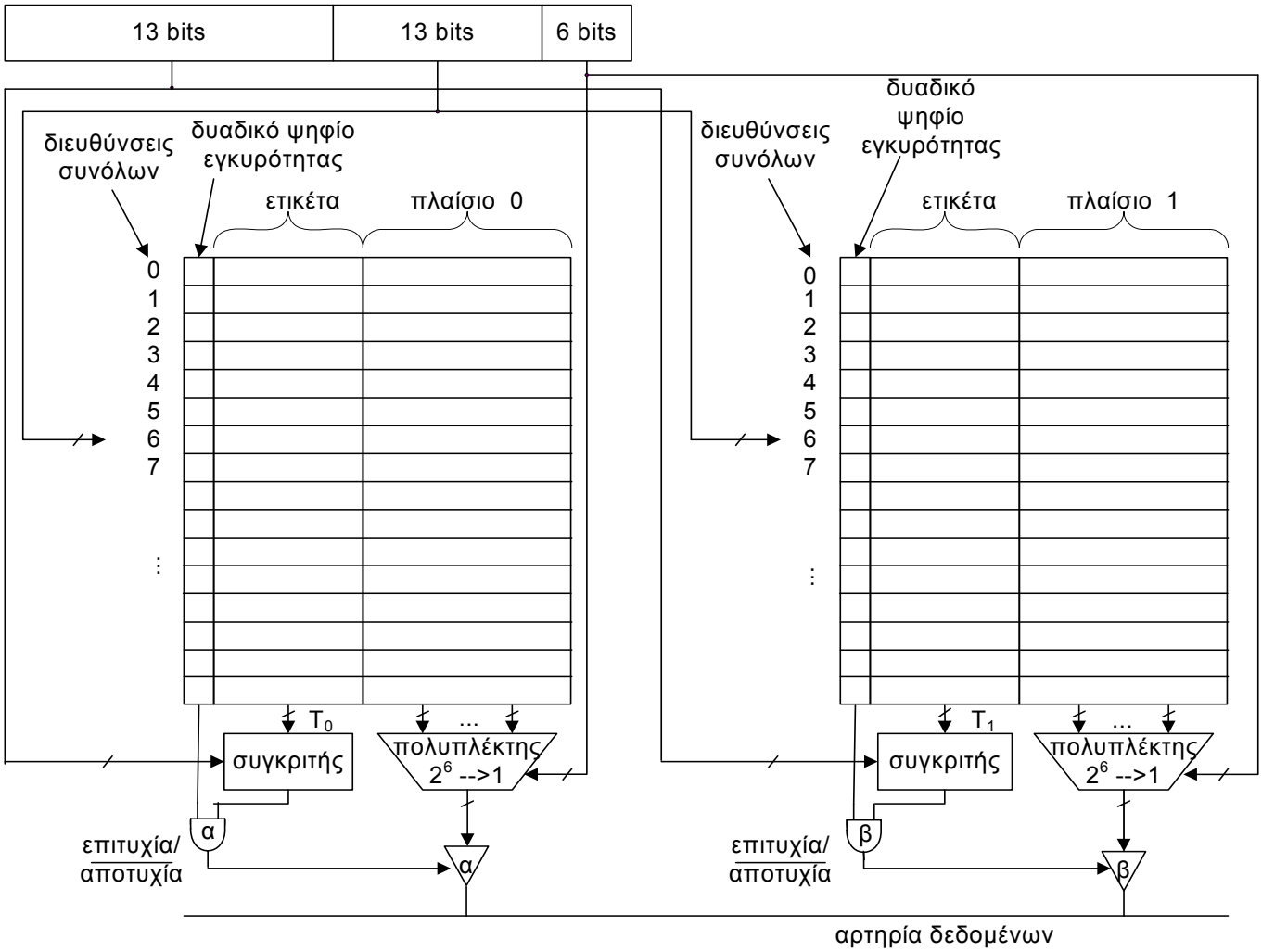
00601F4C(16): διεύθυνση α
0000 0000 0110 0000 0001 1111 0100 1100

00681F4E(16): διεύθυνση β
0000 0000 0110 1000 0001 1111 0100 1110

00601F4F(16): διεύθυνση γ
0000 0000 0110 0000 0001 1111 0100 1111

Οι διευθύνσεις α και γ ανήκουν στο ίδιο μπλοκ της κύριας μνήμης (έχουν την ίδια διεύθυνση συνόλου και την ίδια ετικέτα), άρα όταν μεταφέρεται το περιεχόμενο του μπλοκ στην κρυφή μνήμη θα βρίσκονται σ' αυτή τα περιεχόμενα και των δύο διευθύνσεων. Η διεύθυνση β έχει διαφορετική διεύθυνση μπλοκ από τις προηγούμενες αλλά έχει την ίδια διεύθυνση συνόλου και διαφορετική ετικέτα από τις προηγούμενες. Επομένως αντιστοιχεί

στο ίδιο σύνολο, αλλά αφού κάθε σύνολο έχει δύο πλαίσια μπορεί να βρίσκεται το περιεχόμενό της ταυτόχρονα με τα περιεχόμενα των άλλων διευθύνσεων στην κρυφή μνήμη.



Θέμα 4. Ποιά είναι η κύρια διαφορά μεταξύ ενός ή υπερβαθμωτού (superscalar) επεξεργαστή και ενός επεξεργαστή πολύ μεγάλου μήκους εντολών (Very Long Instruction Word, VLIW); (Μονάδες: 2)

Απάντηση.
Σ' ένα επεξεργαστή πολύ μεγάλου μήκους εντολών, ΠΜΜΕ, η επιλογή των λειτουργιών που θα σταλούν για εκτέλεση σε ένα κύκλο γίνεται κατά τη μεταγλώττιση του πηγαίου προγράμματος. Αυτό καλείται στατικός χρονοπρογραμματισμός (static scheduling). Αντίθετα σε ένα υπερβαθμωτό επεξεργαστή ο χρονοπρογραμματισμός γίνεται από κυκλώματα κατά την εκτέλεση του εκτελέσιμου προγράμματος και γι' αυτό καλείται δυναμικός χρονοπρογραμματισμός (dynamic scheduling).