

Πανεπιστήμιο Θεσσαλίας

Τμήμα Πληροφορικής

Οργάνωση Η/Υ

Ενότητα 3η: Μονάδα Ελέγχου Πολλαπλών Κύκλων Μηχανής

Άσκηση 1:

Θεωρήστε τη Μονάδα Επεξεργασίας Δεδομένων της απλοποιημένης αρχιτεκτονικής MIPS, στην οποία κάθε κύκλος εντολής καταλαμβάνει πολλαπλούς κύκλους μηχανής. Η ροή πληροφορίας που είδαμε στο μάθημα δεν καλύπτει τις παρακάτω περιπτώσεις που σας ζητείται να αναλύσετε:

(α) Εντολές ΑΛΜ με άμεσο τελούμενο.

(β) Εντολές ΑΛΜ με μία έμμεση αναφορά στη μνήμη, τόσο χωρίς όσο και με μετατόπιση.

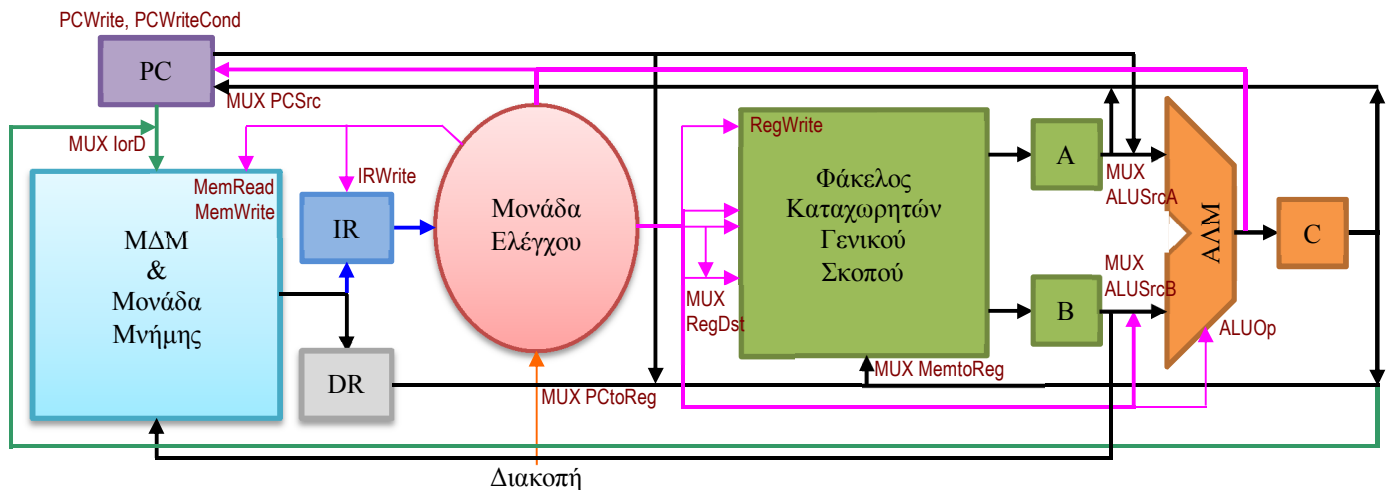
(γ) Εντολές άλματος με σύνδεση.

Περιγράψτε τη ροή πληροφορίας για κάθε περίπτωση και δείξτε τα αντίστοιχα συμπληρώματα για τη μηχανή καταστάσεων.

Όπου θεωρήσετε απαραίτητο, τροποποιήστε την αρχιτεκτονική με νέους δρόμους πληροφορίας και νέους ή διευρυμένους πολυπλέκτες επιλογής πληροφορίας.

Απάντηση:

Το διάγραμμα της Μονάδας Επεξεργασίας Δεδομένων (ΜΕΔ) που μελετήσαμε στο μάθημα είναι το εξής:



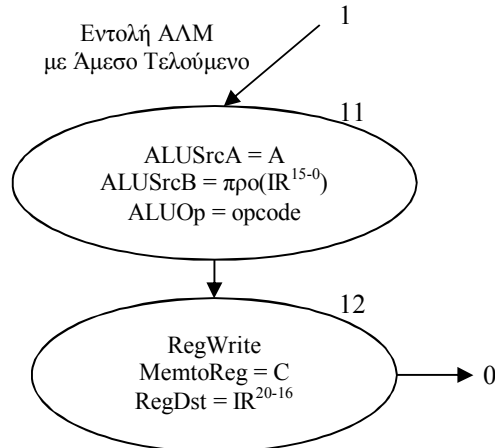
(α) Στο παραπάνω διάγραμμα παρατηρούμε ότι υπάρχει υποστήριξη για ροή πληροφορίας άμεσου τελούμενου προς την ΑΛΜ, την οποία ούτως ή άλλως χρησιμοποιήσαμε τόσο στις εντολές άλματος με συνθήκη (σχετικό άλμα) όσο και στις εντολές προσπέλασης της μνήμης. Έτσι, σε μια εντολή ΑΛΜ με άμεσο τελούμενο η φάση εκτέλεσης γίνεται ως εξής:

$$C = A \text{ op προέκταση}(\text{IR}^{15-0})$$

Για απλούστευση δεν θα ασχοληθούμε με την επιλογή προέκτασης μηδενικών για λογικές εντολές. Όσο αφορά τη φάση αποθήκευσης αποτελέσματος, αυτή γίνεται στον καταχωρητή με διεύθυνση που καθορίζεται από τα ψηφία IR^{20-16} , επιλογή που έχουμε ήδη κάνει στις εντολές ανάγνωσης από τη μνήμη:

$$\text{REG}[\text{IR}^{20-16}] = C$$

Η άμεση διευθυνσιοδότηση σε εντολές ΑΛΜ προσθέτει δύο καταστάσεις στη μηχανή καταστάσεων που είδαμε, οι οποίες αναπαριστώνται όπως φαίνεται πιο κάτω, ενώ η μετάβαση από την κατάσταση 1 στην κατάσταση 6 τώρα γίνεται, μόνο αν η εντολή είναι εντολή ΑΛΜ χωρίς άμεσο τελούμενο:



Όπως βλέπουμε στις νέες καταστάσεις 11 και 12, τόσο οι επιλογές εισόδων της ΑΛΜ, όσο και τα σήματα που σχετίζονται με την αποθήκευση αποτελέσματος στο ΦΚ, απαντώνται σε ήδη υπάρχουσες καταστάσεις. Από την άλλη μεριά, η μικρολειτουργία πράξης στην κατάσταση 11 καθορίζεται από τον κωδικό λειτουργίας (opcode), κάτι που να μην δεν υπάρχει σε άλλη κατάσταση, μπορεί όμως να υλοποιηθεί εύκολα στην ΑΛΜ, μια που όπως έχουμε δει σε προηγούμενο μάθημα, το σήμα επιλογής πράξης μπορεί να παραχθεί ως λογική συνάρτηση του κωδικού λειτουργίας και του κωδικού τελεστή. Έτσι, με το σήμα “opcode” καλύπτουμε όλες τις εντολές ΑΛΜ με άμεσο τελούμενο. Εναλλακτικά, θα έπρεπε να έχουμε ξεχωριστή κατάσταση για κάθε πράξη ΑΛΜ που θέλουμε να υποστηρίξουμε, με το σήμα ελέγχου της ΑΛΜ να δίνει απ’ ευθείας την αντίστοιχη μικρολειτουργία (πχ. add, or, κλπ). Επομένως πέρα από την επιλογή πράξης ΑΛΜ, η μόνο άλλη απαιτούμενη παρέμβαση στο υλικό είναι η τροποποίηση στο κύκλωμα ελέγχου με την προσθήκη των νέων καταστάσεων.

(β) Όπως είδαμε σε άσκηση της προηγούμενης ενότητας, για να υποστηρίξουμε έμμεση αναφορά στη μνήμη χωρίς μετατόπιση σε εντολή ΑΛΜ, πρέπει να υποστηρίξουμε ροή πληροφορίας από το φάκελο καταχωρητών γενικού σκοπού προς τη ΜΔΜ (είσοδο διευθύνσεων) και αντίστοιχη ροή από τη ΜΔΜ (έξοδος δεδομένων) προς την ΑΛΜ. Στην παρούσα ενότητα, που έχουμε καταχωρητές ειδικού σκοπού για την προσωρινή αποθήκευση ενδιάμεσων αποτελεσμάτων στον κύκλο εντολής, οι αντίστοιχες ροές θα είναι από τον καταχωρητή A προς την είσοδο διευθύνσεων της ΜΔΜ και από τον καταχωρητή DR προς την ΑΛΜ. Και για τις δύο αυτές ροές χρειαζόμαστε τη δυνατότητα επιπρόσθετης επιλογής στις αντίστοιχες εισόδους: Επιλογή στην είσοδο διευθύνσεων της ΜΔΜ από μονάδα PC, από C και από A, και επιλογή στην είσοδο ΑΛΜ1 από PC, από A και από DR. Οι σχετικές τροποποιήσεις της ΜΕΔ φαίνονται στο νέο διάγραμμα της επόμενης σελίδας.

Ο κύκλος εντολής στην περίπτωση εντολής ΑΛΜ με έμμεση αναφορά στη μνήμη χωρίς μετατόπιση πρέπει να περνάει μια φάση προσπέλασης στη μνήμη πριν τη φάση εκτέλεσης:

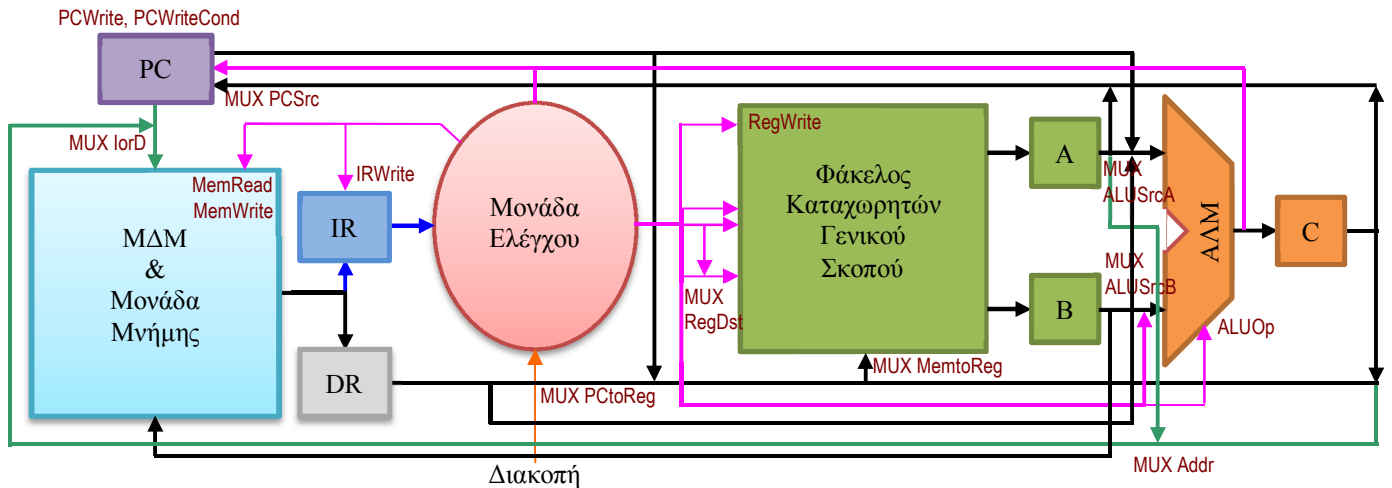
$$DR = MEM[A]$$

με τη φάση εκτέλεσης να γίνεται:

$$C = DR \text{ or } B$$

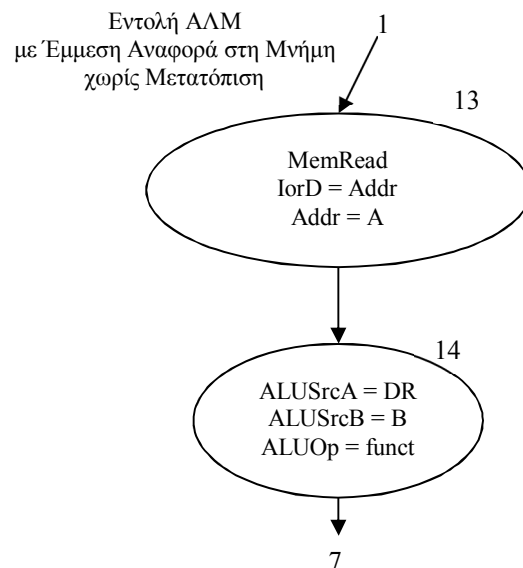
και την ίδια όπως και πρώτα φάση αποθήκευσης αποτελέσματος.

Η παραπάνω διαδοχή φάσεων θα αναπαρασταθεί με δύο πρόσθετες καταστάσεις 13 και 14. Η επιλογή στην είσοδο ΑΛΜ1 αναγράφεται όπως και πρώτα, υπονοώντας βέβαια διεύρυνση του πολυπλέκτη ALUSrcA σε 4×1 . Η επιλογή στην είσοδο διευθύνσεων της ΜΔΜ μπορεί επίσης να γίνει με διεύρυνση του πολυπλέκτη IorD, αλλά εδώ επιλέξαμε την υλοποίηση με δύο διαδοχικούς πολυπλέκτες 2×1 , τον IorD και ένα νέο πολυπλέκτη Addr που επιλέγει με-



ταξύ A και C, και ο οποίος τροφοδοτεί τον πρώτο στην επιλογή του για διεύθυνση δεδομένων. Το σήμα που ελέγχει το νέο πολυπλέκτη θα πρέπει να χρησιμοποιείται σε όλες τις καταστάσεις που ξεκινούν προσπέλαση στη μνήμη για δεδομένα, άρα και στις καταστάσεις 3 και 5, στις οποίες θα έχουμε τις επιλογές “IorD = Addr” και “Addr = C”. Στην κατάσταση 0 που ξεκινά προσπέλαση στη μνήμη για εντολή, το σήμα αυτό μας είναι αδιάφορο, γιατί η επιλογή “IorD = PC” προσδιορίζει την είσοδο διευθύνσεων της ΜΔΜ από τη μονάδα PC και ο νέος πολυπλέκτης δε χρησιμοποιείται.

Σχηματικά, οι δύο πρόσθετες καταστάσεις απεικονίζονται στο διάγραμμα που ακολουθεί, όπου βέβαια η υπάρχουσα μετάβαση από την κατάσταση 1 στην κατάσταση 6, όπως και η μετάβαση από την κατάσταση 1 στην κατάσταση 11 που εισάγαμε νωρίτερα, γίνεται όταν η εντολή ΑΛΜ δεν έχει αναφορά στη μνήμη:



Πέρα από τις τροποποιήσεις της ΜΕΔ που δείξαμε νωρίτερα, η προσθήκη των δύο νέων καταστάσεων συνοδεύεται από αντίστοιχη τροποποίηση του κυκλώματος ελέγχου.

Πριν προχωρήσουμε στις εντολές ΑΛΜ με έμμεση αναφορά στη μνήμη με μετατόπιση, μπορούμε να παρατηρήσουμε ότι η παραπάνω περίπτωση επεκτείνεται για άμεση διευθυνσιοδότηση στην πράξη ΑΛΜ με τον ίδιο τρόπο όπως στην περίπτωση (α), εισάγοντας μια νέα κατάσταση 15. Μπορούμε δε στην περίπτωση αυτή να ξαναχρησιμοποιήσουμε την κατάσταση 12 σαν επόμενη της 15.

Όταν έχουμε μετατόπιση στην αναφορά στη μνήμη, η αρχιτεκτονική MIPS μας επιτρέπει να χρησιμοποιούμε μέχρι δύο καταχωρητές-τελούμενα. Άρα για να υποστηρίξουμε μετατόπιση στην αναφορά στη μνήμη σε εντολή ΑΛΜ, πρέπει να υποθέσουμε ότι ο ένας τελεστέος ταυτίζεται με τον καταχωρητή-αποτέλεσμα της πράξης.

Όπως και παραπάνω, μια φάση ανάγνωσης από τη μνήμη θα προηγείται της φάσης εκτέλεσης στην ΑΛΜ. Η ύπαρξη μετατόπισης όμως, μας αναγκάζει να εισάγουμε μία προηγούμενη φάση εκτέλεσης πριν από τη φάση ανάγνωσης του τελεστέου από τη μνήμη:

$$C = A + \text{προέκταση}(\text{IR}^{15-0})$$

Η διεύθυνση τώρα θα διαβαστεί από τον καταχωρητή C, το οποίο σημαίνει ότι δε χρειαζόμαστε στην περίπτωση αυτή το δρόμο πληροφορίας από τον A στη ΜΔΜ:

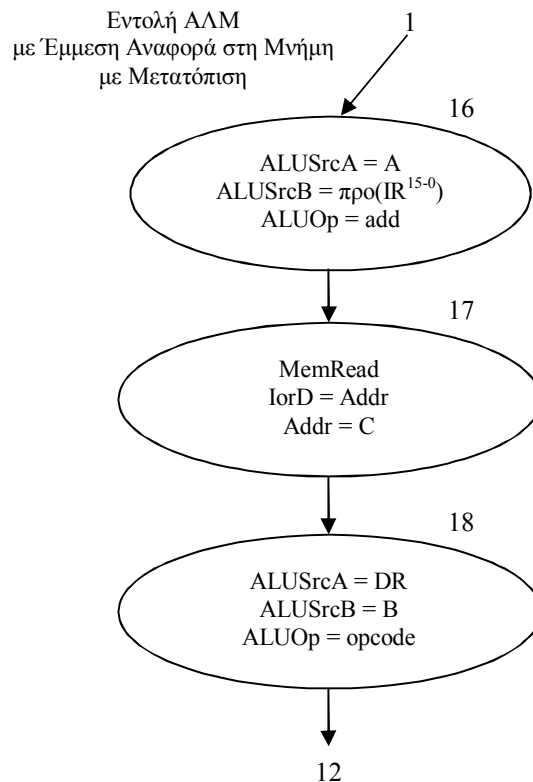
$$\text{DR} = \text{MEM}[C]$$

και η δεύτερη φάση εκτέλεσης θα είναι πάλι:

$$C = \text{DR} \text{ op } B$$

Όμως τώρα η πράξη op δε μπορεί να καθοριστεί από τον κωδικό τελεστή funct, εφόσον τα ψηφία IR^{5-0} , τα οποία αντιστοιχούν στον κωδικό τελεστή σε εντολές τύπου R, εδώ χρησιμοποιούνται για τη μετατόπιση. Οπότε, όπως ήδη εξηγήσαμε στην περίπτωση (α), η πράξη θα καθορίζεται από τον κωδικό λειτουργίας opcode. Η φάση αποθήκευσης αποτελέσματος θα είναι η ίδια με την περίπτωση (α), μια που η διεύθυνση του καταχωρητή αποτελέσματος θα καθορίζεται από τα ψηφία IR^{20-16} . Για το λόγο αυτό μπορούμε να χρησιμοποιήσουμε την κατάσταση 12 της περίπτωσης (α) ως την τελευταία κατάσταση των εντολών που μελετάμε.

Εισάγουμε έτσι ακόμα τρεις καταστάσεις 16, 17 και 18, όπως φαίνεται στο παρακάτω διάγραμμα:



Οι μικρολειτουργίες και επιλογές που αναγράφονται στις νέες καταστάσεις υπάρχουν ήδη σε άλλες καταστάσεις, κι επομένως δε χρειαζόμαστε καμία περαιτέρω προσθήκη σε υλικό εκτός από τις τροποποιήσεις στο κύκλωμα ελέγχου.

Εναλλακτικά, αν παρατηρήσουμε προσεκτικά τη μηχανή καταστάσεων, θα δούμε ότι οι καταστάσεις 16 και 17 ενεργοποιούν τις ίδιες μικρολειτουργίες με τις καταστάσεις 2 και 3, ενώ τόσο η 16 όσο και η 2 ακολουθούν την 1. Έτσι, με κατάλληλη συνθήκη στην έξοδο από την 3 μπορούμε να οδηγηθούμε είτε στην 4 είτε στην 18, απαλείφοντας έτσι την εισαγωγή των δύο

από τις τρεις νέες καταστάσεις για την υποστήριξη εντολών ΑΛΜ με έμμεση αναφορά στη μνήμη με μετατόπιση.

(γ) Για την εκτέλεση άλματος με σύνδεση, τόσο για άμεσο όσο και για έμμεσο άλμα, χρησιμοποιούμε το δρόμο πληροφορίας από τη μονάδα PC στο φάκελο καταχωρητών. Το πρόβλημα που υπάρχει με τη σύνδεση στη διεύθυνση επανόδου είναι ότι η αποθήκευση της τιμής του μετρητή προγράμματος στο φάκελο καταχωρητών δε μπορεί να γίνει μετά την παραλαβή της διεύθυνσης προορισμού άλματος από το μετρητή προγράμματος. Άρα η φάση αποθήκευσης στο φάκελο καταχωρητών πρέπει στην περίπτωση αυτή να γίνει πριν ή ταυτόχρονα με τη φάση εκτέλεσης. Εφ' όσον η αποθήκευση στον PC – όπως και κάθε αποθήκευση – τελείται με την πτώση του παλμού μηχανής, η προηγούμενη τιμή του PC είναι διαθέσιμη για ανάγνωση καθ' όλη τη διάρκεια του ίδιου παλμού μηχανής πριν την πτώση του στο επίπεδο 0 ή σε οποιονδήποτε προηγούμενο κύκλο μηχανής που ακολουθεί την τελευταία εγγραφή στον PC.

Στο άμεσο άλμα το αποτέλεσμα εγγράφεται στον καταχωρητή 31, οπότε ενοποιώντας τις δύο φάσεις σύμφωνα με τα παραπάνω, η φάση εκτέλεσης/αποθήκευσης θα είναι:

$$PC = (PC^{31-28} \parallel IR^{25-0}) \ll 2$$

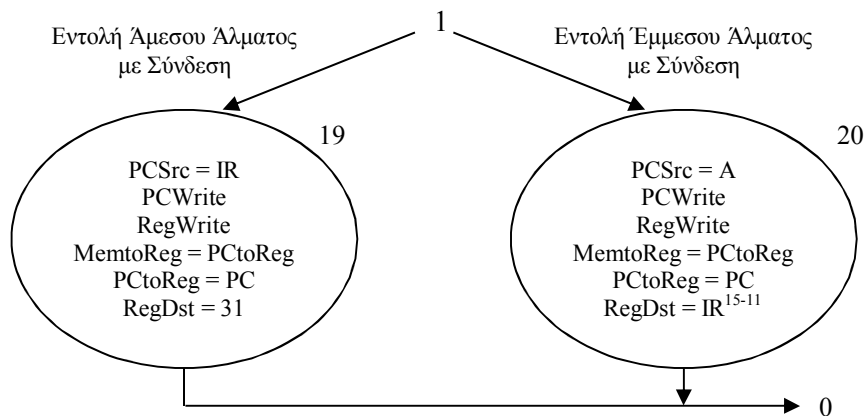
$$REG[31] = PC$$

Στην περίπτωση έμμεσου άλματος το αποτέλεσμα εγγράφεται στον καταχωρητή με διεύθυνση που καθορίζεται από τα ψηφία IR^{15-11} , και η φάση αυτή θα γίνει:

$$PC = A$$

$$REG[IR^{15-11}] = PC$$

Στο παραπάνω διάγραμμα της ΜΕΔ υπάρχει πρόβλεψη για ροή πληροφορίας από τη μονάδα PC προς το φάκελο καταχωρητών. Άρα θεωρούμε δεδομένη τη δυνατότητα επιλογής εισόδου στο φάκελο καταχωρητών από τη μονάδα PC μέσω του πολυπλέκτη PCtoReg. Πέρα από την προσθήκη της σταθεράς 31 στις επιλογές διεύθυνσης αποθήκευσης στο φάκελο καταχωρητών – άρα και επέκταση του αντίστοιχου πολυπλέκτη από 2×1 σε 4×1 , η μόνη άλλη επέμβαση στο υλικό που χρειαζόμαστε είναι η υποστήριξη δύο νέων καταστάσεων από το κύκλωμα ελέγχου, όπως φαίνεται στο πιο κάτω συμπληρωματικό διάγραμμα:



όπου η επιλογή "PCSrc = IR" συμβολίζει την απόδοση στον μετρητή προγράμματος της τιμής που αναφέρθηκε παραπάνω. Οι καταστάσεις 9 και 10 τώρα πια αναφέρονται σε άλματα χωρίς σύνδεση.

Άσκηση 2:

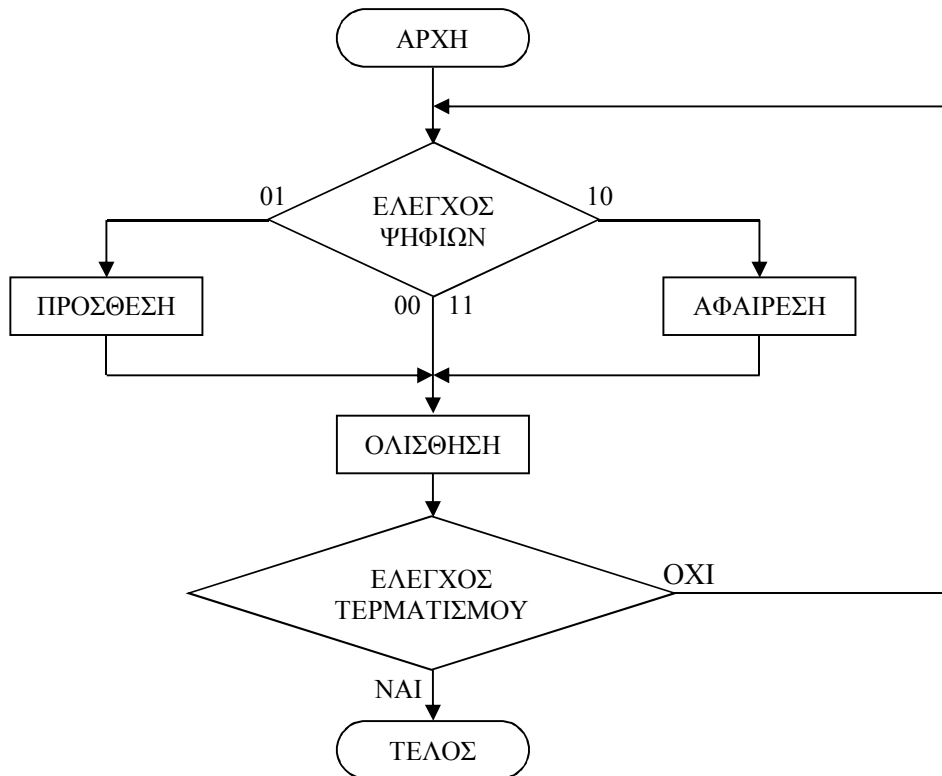
Θεωρήστε τη μονάδα πολλαπλασιασμού ακεραίων με μια ΑΛΜ των 32 bits, έναν καταχωρητή ολίσθησης K (γινόμενο/πολλαπλασιαστή) μήκους 64 bits και έναν καταχωρητή $K1$ (πολλαπλασιαστέο) μήκους 32 bits.

Θεωρήστε τον επαναληπτικό αλγόριθμο πολλαπλασιασμού στηριγμένο στη μέθοδο Booth και τα σήματα ελέγχου «Πράξη/Όχι πράξη» και «Αφαίρεση/Πρόσθεση» για επιλογή πράξης.

Σχεδιάστε την υπομονάδα ελέγχου της μονάδας, υποθέτοντας ότι διαθέτετε τα παραπάνω σήματα. Κάθε μικρολειτουργία εκτελείται σε έναν κύκλο μηχανής. Ο χρονισμός της μονάδας πρέπει να είναι τέτοιος, ώστε να ολοκληρώνει έναν πολλαπλασιασμό στο συντομότερο δυνατό χρόνο. Συμπληρώστε τη μονάδα με όποια στοιχεία θεωρείτε απαραίτητα.

Απάντηση:

Το διάγραμμα του αλγόριθμου πολλαπλασιασμού με βάση τη μέθοδο Booth είναι το παρακάτω:



Για την προσαρμογή του αλγόριθμου αυτού στη μονάδα που διαθέτουμε, πρέπει κατ' αρχήν να παρατηρήσουμε ότι όπως και στον επαναληπτικό αλγόριθμο πολλαπλασιασμού χωρίς τη μέθοδο Booth, έτσι κι εδώ, κάθε πράξη ΑΛΜ γίνεται με το πρώτο μισό του K, ενώ κάθε ολίσθηση γίνεται σε ολόκληρο τον K. Στον K τοποθετούμε αρχικά τον πολλαπλασιαστή, με τρόπο που θα καθορίσουμε πιο κάτω, ώστε ο αλγόριθμος Booth να ελέγχει τα δύο λιγότερο σημαντικά ψηφία του K σε κάθε επανάληψη.

Για τον έλεγχο τερματισμού του αλγόριθμου, εισάγουμε ένα μετρητή N μεγέθους 5 bits, που μετρά από 31 μέχρι 0, οπότε ο έλεγχος τερματισμού θα είναι ο έλεγχος μηδενισμού του N. Ο N αρχικοποιείται στην τιμή 31.

Με την εισαγωγή του μετρητή και των σχετικών με αυτόν μικρολειτουργιών, μπορούμε να δώσουμε σε μορφή πίνακα όλες τις μικρολειτουργίες της μονάδας:

Μικρολειτουργία	Επεξήγηση
initN	Αρχικοποίηση N με τιμή 31
N--	Μείωση N κατά 1
op(Booth)	Αριθμητική πράξη από έλεγχο Booth
rShiftK	Δεξιά ολίσθηση K
writeHalfK	Εγγραφή αριστερού μισού K

Σύμφωνα με την εκφώνηση, στη συγκεκριμένη μονάδα που διαθέτουμε, κάθε μικρολειτουργία απαιτεί έναν κύκλο μηχανής. Έτσι, για την υλοποίηση του ελέγχου¹, πρέπει να διαχωρίσουμε τα σήματα που σχετίζονται με την πράξη ΑΛΜ από εκείνα που σχετίζονται με την ολίσθηση του K , και να τα αντιστοιχήσουμε σε διαφορετικές καταστάσεις της μονάδας. Άλλη κατάσταση επομένως θα ενεργοποιεί το σήμα εκτέλεσης πράξης ΑΛΜ, και άλλη κατάσταση θα ενεργοποιεί το σήμα ολίσθησης του K .

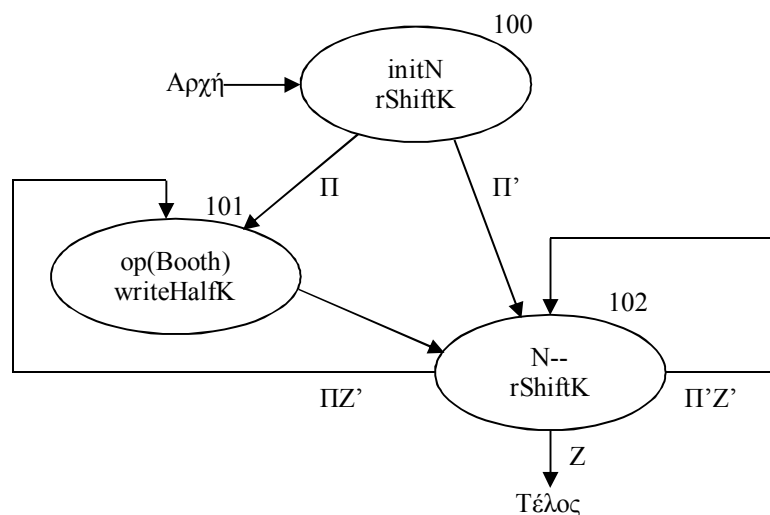
Για να επιτύχουμε δε το συντομότερο χρόνο στην ολοκλήρωση ενός πολλαπλασιασμού, πρέπει να μπορούμε να επανέλθουμε στην ολίσθηση αμέσως μετά την προηγούμενη ολίσθηση, για τις επαναλήψεις στις οποίες η συνθήκη των ψηφίων K_1K_0 του καταχωρητή K αποτρέπει την εκτέλεση πράξης στην ΑΛΜ. Άρα, ο έλεγχος της συνθήκης ψηφίων θα γίνεται στο βήμα ολίσθησης. Όμως, ο έλεγχος αυτός δε μπορεί να χρησιμοποιεί το αποτέλεσμα της ολίσθησης, επειδή τότε η μικρολειτουργία της ολίσθησης, ο έλεγχος Booth, αλλά και ο επακόλουθος υπολογισμός επόμενης κατάστασης, θα έπρεπε να τελούνται στη σειρά και δε θα χωρούσαν σε έναν κύκλο μηχανής. Επομένως, ο έλεγχος θα αναφέρεται αναγκαστικά στον αριθμό που είναι αποθηκευμένος στον K πριν την ολίσθηση, έτσι ώστε να μπορεί να γίνει ταυτόχρονα με αυτή. Υποθέτουμε άνω ότι ο έλεγχος Booth είναι σύντομος, ώστε ο υπολογισμός επόμενης κατάστασης να μπορεί να ακολουθεί στον ίδιο κύκλο.

Σα συνέπεια των παραπάνω, κατά την αρχικοποίηση των καταχωρητών K και $K1$, ο πολλαπλασιαστής τοποθετείται στα ψηφία K^{32-1} , ενώ το ψηφίο K_0 λαμβάνει τιμή 0. Έτσι στην ουσία το υπονοούμενο ψηφίο του αλγόριθμου Booth υλοποιείται στο λιγότερο σημαντικό ψηφίο του K .

Το αποτέλεσμα του ελέγχου χρησιμοποιείται για τη εύρεση επόμενης κατάστασης, αλλά και για την επιλογή πράξης ΑΛΜ στον επόμενο κύκλο. Για να διατηρούμε το αποτέλεσμα του ελέγχου πέραν του ενός κύκλου, εισάγουμε και ένα ψηφίο κατάστασης (απλό D flip-flop – έστω F), όπου καταχωρείται το σήμα “Αφαίρεση/Πρόσθεση” στο τέλος του κύκλου.

Η μείωση του μετρητή N κατά 1 γίνεται στο βήμα ολίσθησης, και ο υπολογισμός της συνθήκης τερματισμού γίνεται στην τιμή του N πριν τη μείωση, ώστε να πάρουμε ακριβώς 32 επαναλήψεις.

Η μηχανή καταστάσεων της μονάδας πολλαπλασιασμού θα περιλαμβάνει τρεις καταστάσεις. Μία αρχική κατάσταση, όπου με την έναρξη του αλγόριθμου αρχικοποιείται ο μετρητής N , γίνεται ο πρώτος έλεγχος Booth και τελείται η πρώτη ολίσθηση του K , και δύο ακόμα καταστάσεις για την πράξη ΑΛΜ και για την ολίσθηση του K πέραν της πρώτης. Έτσι, καταλήγουμε στην παρακάτω μηχανή καταστάσεων:



¹ Υποθέτουμε ότι η μονάδα πολλαπλασιασμού επικοινωνεί με μια εξωτερική μονάδα ελέγχου, η οποία φροντίζει για την τροφοδοσία των K και $K1$, καθώς και για τη λήψη του αποτελέσματος. Η επικοινωνία αυτή δε μας απασχολεί σε αυτή την άσκηση.

Ξεκινώντας από την κατάσταση 100, εκτελούμε μια ολίσθηση, και ανάλογα με την τιμή των αρχικών ψηφίων K_1K_0 μεταβαίνουμε σε μια από τις καταστάσεις 101 και 102. Από την 101, όπου τελείται η πράξη ΑΛΜ, μεταβαίνουμε υποχρεωτικά στην 102, όπου τελείται η ολίσθηση. Από την 102, έχουμε πάλι επιλογή μετάβασης προς τις 101 και 102, ανάλογα με την τιμή των ψηφίων K_1K_0 , ενώ αν ο μετρητής N έχει μηδενιστεί, σηματοδοτείται η ολοκλήρωση του πολλαπλασιασμού.

Παρατηρήστε ότι δε χρειαζόμαστε διαφορετικές καταστάσεις για τις διαφορετικές πράξεις ΑΛΜ, αλλά μόνο μία, αφού, εφ' όσον διαθέτουμε το σήμα “Αφαίρεση/Πρόσθεση” από το ψηφίο κατάστασης F , μπορούμε να ενσωματώσουμε τις μικρολειτουργίες “πρόσθεση” και “αφαίρεση” σε μία μικρολειτουργία, αυτή που ονομάσαμε “op(Booth)”. Στο διάγραμμα σημειώσαμε ως “Π” το σήμα “Πράξη/Όχι πράξη”, ενώ “Ζ” είναι το σήμα ελέγχου μηδενικής τιμής του N .

Με ένα bit ανά κατάσταση, οι λογικές εκφράσεις επόμενης κατάστασης καθώς και η συνθήκη τερματισμού θα είναι:

$$\begin{aligned} S_{101} &= S_{100}\Pi + S_{102}\Pi Z' = \Pi (S_{100} + S_{102}Z') \\ S_{102} &= S_{100}\Pi' + S_{102}\Pi'Z' + S_{101} = \Pi' (S_{100} + S_{102}Z') + S_{101} \\ \text{Τέλος} &= S_{102}Z \end{aligned}$$

Στην ουσία η συνθήκη τερματισμού αποτελεί σήμα που στέλνεται στην κεντρική ΜΕ της ΜΕΔ για σηματοδότηση της ολοκλήρωσης της εκτέλεσης μιας πράξης πολλαπλασιασμού.

Τα σήματα ελέγχου της μονάδας δίνονται από τις εκφράσεις:

$$\begin{aligned} \text{“rShiftK”} &= S_{100} + S_{102} \\ \text{“writeHalfK”} &= S_{101} \\ \text{“initN”} &= S_{100} \\ \text{“N--”} &= S_{102} \end{aligned}$$

ενώ το σήμα επιλογής τελεστή στην ΑΛΜ θα είναι σύμφωνα με τα παραπάνω:

$$\text{“op(Booth)”} = F$$

Παρατηρούμε ότι με την πιο πάνω υλοποίηση και τους δεδομένους περιορισμούς υλικού, ο χρόνος ολοκλήρωσης ενός πολλαπλασιασμού είναι πράγματι ελάχιστος, εφ' όσον ο αριθμός κύκλων μηχανής που απαιτεί είναι 32 συν τον αριθμό των πράξεων ΑΛΜ που εκτελούνται.

Άσκηση 3:

Θεωρήστε κάποια αρχιτεκτονική συσσωρευτή με τις εξής εντολές και μεθόδους διευθυνσιοδότησης:

Διευθυνσιοδότηση	Εντολή							
	LOAD	STORE	ADD	SUB	JMP	JAL	BNZ	BLTZ
Κατ' ευθείαν	1	1	1	1	1	1	0	0
Άμεση	1	0	1	0	0	0	0	0
Έμμεση	1	1	1	1	1	1	0	0
Σχετική	0	0	0	0	0	0	1	1

όπου οι εντολές *LOAD* και *STORE* φορτώνουν στον και αποθηκεύουν από το συσσωρευτή αντίστοιχα, οι εντολές *ADD* και *SUB* προσθέτουν στον και αφαιρούν από το συσσωρευτή αντίστοιχα, η εντολή *JMP* εκτελεί άλμα, η εντολή *JAL* εκτελεί άλμα με σύνδεση στο συσσωρευτή, ενώ οι εντολές *BNZ* και *BLTZ* εκτελούν άλμα με συνθήκη μη μηδενική και αρνητική τιμή του συσσωρευτή αντίστοιχα.

Το μέγεθος λέξης της αρχιτεκτονικής είναι 32bits. Σε μια λέξη εντολής, τα 3 πιο σημαντικά ψηφία περιέχουν τον κωδικό λειτουργίας, τα 4 επόμενα τη μέθοδο διευθυνσιοδότησης (ένα ψηφίο ανά μέθοδο), και τα υπόλοιπα το τελούμενο. Αν αυτό είναι διεύθυνση, τα 7 πιο σημαντικά ψηφία του λαμβάνονται από το μετρητή προγράμματος. Διαφορετικά, προέκταση προσήμου δίνει αριθμό (δεδομένο ή μετατόπιση) μεγέθους 32 bits. Επειδή όλα τα δεδομένα που αναφέρονται είναι μεγέθους λέξης, η μνήμη διευθυνσιοδοτείται σε λέξεις.

Προσαρμόζοντας την αρχιτεκτονική MIPS σε αρχιτεκτονική συσσωρευτή, δώστε μια Μονάδα Επεξεργασίας Δεδομένων (ΜΕΔ) και σχεδιάστε μια Μονάδα Ελέγχου (ΜΕ) πολλαπλών κύκλων μηχανής ανά κύκλο εντολής για την παραπάνω αρχιτεκτονική.

Απάντηση:

Για να μπορέσουμε να μετατρέψουμε τη γνωστή μας ΜΕΔ της αρχιτεκτονικής MIPS στη ζητούμενη, θα πρέπει πρώτα να εξετάσουμε πώς εκτελούνται οι εντολές μιας αρχιτεκτονικής συσσωρευτή.

Κατ' αρχήν, ανεξάρτητα από την αρχιτεκτονική, σε οποιονδήποτε επεξεργαστή ακολουθεί το μοντέλο εκτέλεσης αποθηκευμένου προγράμματος, γνωστό και ως *μοντέλο von Neumann*, ο κύκλος εντολής περνάει από τις φάσεις ανάκλησης και αποκωδικοποίησης. Επομένως, η ΜΕΔ που σχεδιάζουμε δε θα διαφέρει από τη ΜΕΔ MIPS στις υπομονάδες που σχετίζονται με τις δύο αυτές φάσεις του κύκλου εντολής. Ας δούμε λοιπόν τι συμβαίνει στη συνέχεια.

Σε μια αρχιτεκτονική συσσωρευτή, οι περισσότερες εντολές έχουν το συσσωρευτή σαν υπονοούμενο τελούμενο, καθώς και ένα τελούμενο που φορτώνεται από τη μνήμη. Έτσι, σε αντίθεση με την αρχιτεκτονική MIPS, η οποία, σαν αρχιτεκτονική φόρτωσης-αποθήκευσης, διαβάσει τα τελούμενά της από το φάκελο καταχωρητών, η αρχιτεκτονική συσσωρευτή χρειάζεται μια φάση προσπέλασης της μνήμης αναγκαστικά να προηγείται της φάσης εκτέλεσης της εντολής. Επειδή ακόμα η αρχιτεκτονική συσσωρευτή υποστηρίζει πολλές μεθόδους διευθυνσιοδότησης, της φάσης προσπέλασης μνήμης πρέπει να προηγείται μια φάση εύρεσης της τελικής διεύθυνσης τελούμενου, που στην περίπτωσή μας είναι απαραίτητη για την έμμεση διευθυνσιοδότηση. Η φάση εκτέλεσης είναι και η τελευταία φάση του κύκλου εντολής, αφού λόγω απουσίας φακέλου καταχωρητών, δεν απαιτείται φάση αποθήκευσης αποτελέσματος, και το αποτέλεσμα αποθηκεύεται στο συσσωρευτή στο τέλος της φάσης εκτέλεσης.

Συνολικά λοιπόν, ο κύκλος εντολής μιας αρχιτεκτονικής συσσωρευτή χωρίζεται στις εξής φάσεις: 1.Ανάκληση, 2.Αποκωδικοποίηση, 3.Εύρεση τελικής διεύθυνσης τελούμενου, 4.Προσπέλαση μνήμης, και 5.Εκτέλεση.

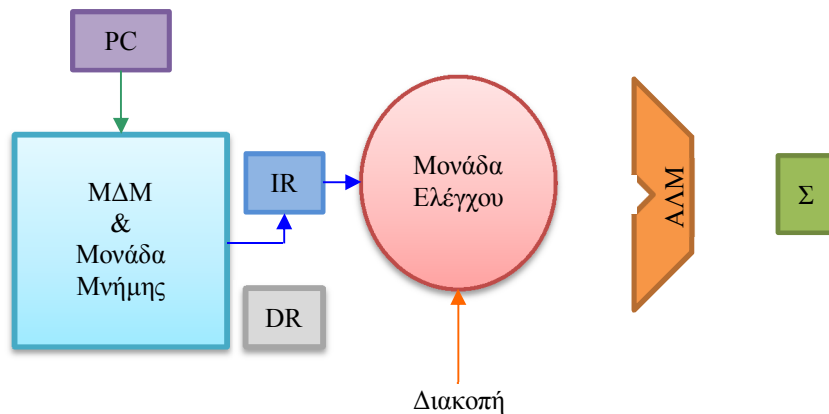
Για να σχεδιάσουμε μια ΜΕΔ σύμφωνη με τις προδιαγραφές εντολών που μας δίνονται, ας αναλογιστούμε τι υπομονάδες χρειαζόμαστε, έχοντας κατά νου τη γνωστή μας ΜΕΔ της αρχιτεκτονικής MIPS.

Έτσι, χρειαζόμαστε κατ' αρχήν μια υπομονάδα ΜΑΜ με μια μνήμη εντολών, μια υπομονάδα ΡC, η οποία να περιέχει το μετρητή προγράμματος ΡC και τη λογική τροποποίησης του περιεχομένου του, και έναν καταχωρητή ειδικού σκοπού ΙR, ο οποίος να αποθηκεύει μια λέξη εντολής.

Επίσης, για υποστήριξη των εντολών που μας δίνονται, χρειαζόμαστε τουλάχιστον μια μνήμη δεδομένων, η οποία για υλοποίηση ΜΕ πολλαπλών κύκλων μηχανής ανά κύκλο εντολής ενσωματώνεται στη μνήμη εντολών, έναν καταχωρητή ειδικού σκοπού DR, ο οποίος να αποθηκεύει μια λέξη δεδομένων, και μια ΑΜΜ.

Για να προσαρμόσουμε τέλος τη ΜΕΔ MIPS σε αρχιτεκτονική συσσωρευτή, χρειαζόμαστε ένα μοναδικό καταχωρητή γενικού σκοπού, το συσσωρευτή Σ, σε αντικατάσταση του φακέλου καταχωρητών.

Το διάγραμμα της ΜΕΔ της αρχιτεκτονικής συσσωρευτή, χωρίς να έχουμε ακόμα εισάγει τις συνδέσεις μεταξύ των υπομονάδων, εκτός από τις προφανείς που σχετίζονται με την ανάκληση και αποκωδικοποίηση των εντολών, θα είναι όπως δίνεται στη συνέχεια, όπου έχουμε ακόμα τοποθετήσει και τη ΜΕ δίπλα στον ΙR, κάτι που κάναμε και στην αρχιτεκτονική MIPS. Για τις συγκεκριμένες εντολές και διευθυνσιοδοτήσεις που υποστηρίζονται από την αρχιτεκτονική μας, δεν απαιτούνται άλλοι καταχωρητές πέρα από όσους φαίνονται στο διάγραμμα.



Παρά το απλοποιημένο σετ εντολών όμως, οι τέσσερις μέθοδοι διευθυνσιοδότησης δημιουργούν την ανάγκη για πολύπλοκους δρόμους ροής πληροφορίας κι επομένως πολύπλοκες συνδέσεις μεταξύ των υπομονάδων.

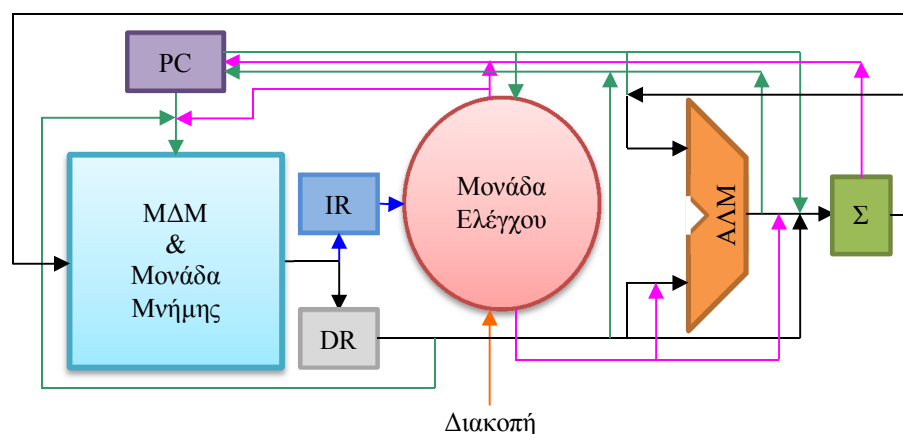
Για να εισάγουμε τις απαιτούμενες συνδέσεις μεταξύ υπομονάδων στο παραπάνω διάγραμμα, ας δούμε πρώτα τι ροή πληροφορίας πρέπει να γίνεται στην εκτέλεση καθεμιάς από τις εντολές, για τις διευθυνσιοδοτήσεις που υποστηρίζουμε:

- Η εντολή LOAD φορτώνει κάποιο δεδομένο στο συσσωρευτή Σ. Με άμεση διευθυνσιοδότηση, το δεδομένο παρέχεται από τη λέξη εντολής. Με κατ' ευθείαν διευθυνσιοδότηση, το δεδομένο προέρχεται από τη μνήμη, από τη διεύθυνση που καθορίζει η λέξη εντολής σε συνδυασμό με τον PC. Με έμμεση διευθυνσιοδότηση, το δεδομένο προέρχεται από τη μνήμη, από διεύθυνση που διαβάζεται από αυτή, από τη διεύθυνση που καθορίζουν η λέξη εντολής και ο PC.
- Η εντολή STORE αποθηκεύει στη μνήμη το περιεχόμενο του Σ. Με κατ' ευθείαν διευθυνσιοδότηση, η διεύθυνση αποθήκευσης καθορίζεται από τη λέξη εντολής σε συνδυασμό με τον PC. Με έμμεση διευθυνσιοδότηση, η διεύθυνση αποθήκευσης διαβάζεται από τη μνήμη, από τη διεύθυνση που καθορίζουν η λέξη εντολής και ο PC.
- Η εντολή ADD προσθέτει κάποιο δεδομένο στον Σ. Με άμεση διευθυνσιοδότηση, το δεδομένο παρέχεται από τη λέξη εντολής. Με κατ' ευθείαν διευθυνσιοδότηση, το δεδομένο προέρχεται από τη μνήμη, από τη διεύθυνση που καθορίζουν η λέξη εντολής και ο PC. Με έμμεση διευθυνσιοδότηση, το δεδομένο προέρχεται από τη μνήμη, από διεύθυνση που διαβάζεται από αυτή, από τη διεύθυνση που καθορίζουν η λέξη εντολής και ο PC.
- Η εντολή SUB αφαιρεί κάποιο δεδομένο από τον Σ. Η ροή πληροφορίας είναι η ίδια, όπως για την εντολή ADD, για τις δύο διευθυνσιοδοτήσεις που υποστηρίζονται.
- Η εντολή JMP εκτελεί άλμα προς κάποια διεύθυνση του κώδικα. Με κατ' ευθείαν διευθυνσιοδότηση, η διεύθυνση προορισμού παρέχεται από το συνδυασμό της λέξης εντολής με τον PC. Με έμμεση διευθυνσιοδότηση, η διεύθυνση προορισμού διαβάζεται από τη μνήμη, από τη διεύθυνση που καθορίζουν η λέξη εντολής και ο PC. Και στις δύο περιπτώσεις, η διεύθυνση προορισμού αποθηκεύεται στον PC.
- Η εντολή JAL εκτελεί άλμα με σύνδεση. Η ροή πληροφορίας είναι όπως στην εντολή JMP, με την προσθήκη της μεταφοράς και αποθήκευσης στον Σ της παλιάς τιμής του PC.
- Η εντολή BNZ εκτελεί άλμα με συνθήκη μη μηδενική τιμή του Σ. Η σχετική διευθυνσιοδότηση επιβάλλει υπολογισμό της διεύθυνσης προορισμού ως το άθροισμα της τιμής του PC με τη μετατόπιση που παρέχεται από τη λέξη εντολής. Ο υπολογισμός αυτός γίνεται στην ΑΛΜ, και το αποτέλεσμα μεταφέρεται στον PC, αν ο Σ περιέχει μη μηδενική τιμή.
- Η εντολή BLTZ εκτελεί άλμα με συνθήκη αρνητική τιμή του Σ. Η ροή πληροφορίας είναι η ίδια, όπως στην εντολή BNZ, με μόνη διαφορά στη συνθήκη που χρησιμοποιείται για τη μεταφορά της διεύθυνσης προορισμού στον PC.

Με βάση τα παραπάνω μπορούμε να μελετήσουμε το ρόλο κάθε καταχωρητή της αρχιτεκτονικής μας:

- Ο καταχωρητής IR περιέχει τη λέξη εντολής που διαβάζεται από τη μνήμη στη φάση ανάκλησης. Το περιεχόμενό του χρησιμοποιείται από τη ΜΕ, τόσο για την αποκωδικοποίηση της εντολής, όσο και για την περαιτέρω προώθηση τμημάτων της λέξης εντολής σε άλλες υπομονάδες της ΜΕΔ. Κάτι τέτοιο είναι απαραίτητο, όταν έχουμε άμεση διευθυνσιοδότηση, για την προώθηση – μετά από προέκταση προσήμου – του άμεσου τελούμενου στον Σ για την εντολή LOAD και στην ΑΛΜ για την εντολή ADD. Επίσης είναι απαραίτητο, όταν έχουμε κατ' ευθείαν διευθυνσιοδότηση στις εντολές JMP και JAL, για την προώθηση της διεύθυνσης προορισμού άλματος στον PC, καθώς και όταν έχουμε σχετική διευθυνσιοδότηση στις εντολές BNZ και BLTZ, για την προώθηση – μετά από προέκταση προσήμου – της μετατόπισης στην ΑΛΜ.
- Ο καταχωρητής DR περιέχει τη λέξη δεδομένων που διαβάζεται από τη μνήμη. Ανάγνωση δεδομένου από τη μνήμη όμως δε γίνεται μόνο στη φάση προσπέλασης μνήμης, οπότε διαβάζεται το ζητούμενο τελούμενο, αλλά και στη φάση εύρεσης τελικής διεύθυνσης τελούμενου, όταν έχουμε έμμεση διευθυνσιοδότηση. Στην τελευταία περίπτωση, διαβάζεται η τελική διεύθυνση του τελούμενου, κι επομένως το περιεχόμενο του DR πρέπει να προωθείται πίσω στη ΜΔΜ, και ειδικότερα στην είσοδο διευθύνσεων αυτής. Για τις εντολές ADD και SUB, το περιεχόμενο του DR προωθείται στην ΑΛΜ. Για τις εντολές JMP και JAL με έμμεση διευθυνσιοδότηση, το περιεχόμενο του DR προωθείται στον PC. Τέλος, για την εντολή LOAD το περιεχόμενο του DR προωθείται στον Σ.
- Ο μετρητής προγράμματος PC δέχεται κάποια διεύθυνση, τόσο από τους καταχωρητές IR και DR, όσο και από την ΑΛΜ. Στην τελευταία περίπτωση δέχεται είτε την αυξημένη κατά 1 τιμή του, είτε τη διεύθυνση προορισμού άλματος στις εντολές BNZ και BLTZ, η οποία όπως είπαμε υπολογίζεται ως το άθροισμα της τιμής του PC (διεύθυνση επόμενης εντολής) με την προέκταση προσήμου της μετατόπισης. Επίσης, ο PC δέχεται και ένα σήμα ελέγχου από το συσσωρευτή Σ, που μεταφέρει την τιμή της συνθήκης που ελέγχεται (τιμή συσσωρευτή είτε διάφορη είτε μικρότερη του 0), για τις ίδιες εντολές. Ο PC στέλνει την τιμή του στη ΜΔΜ για ανάγνωση εντολής, στο συσσωρευτή Σ για υλοποίηση σύνδεσης στην εντολή JAL, στην ΑΛΜ για τον υπολογισμό τόσο της αυξημένης τιμής του όσο και της διεύθυνσης προορισμού άλματος στις εντολές BNZ και BLTZ, ενώ τα 7 πιο σημαντικά ψηφία του συμμετέχουν στον υπολογισμό της διεύθυνσης προορισμού άλματος στις εντολές JMP και JAL με κατ' ευθείαν διευθυνσιοδότηση.
- Ο συσσωρευτής Σ που είναι ο μόνος καταχωρητής άμεσα προσπελάσιμος από κάποιο πρόγραμμα σε γλώσσα μηχανής της αρχιτεκτονικής, δέχεται τιμές από την ΑΛΜ και από όλους τους υπόλοιπους καταχωρητές. Το περιεχόμενό του το στέλνει στην ΑΛΜ και στη ΜΔΜ για αποθήκευση σε εντολή STORE. Στέλνει δε και στον PC την πληροφορία για μη μηδενική ή αρνητική τιμή όπως ήδη προαναφέραμε. Μπορούμε να υποθέσουμε ότι για το σκοπό αυτό παρέχονται δύο ψηφία ελέγχου της τιμής του Σ, το Z για μηδενική και το N για αρνητική τιμή.

Μπορούμε τώρα να συμπληρώσουμε το διάγραμμα της ΜΕΔ με τις απαιτούμενες συνδέσεις μεταξύ των υπομονάδων, καταλήγοντας στο παρακάτω διάγραμμα:



όπου οι δρόμοι ροής πληροφορίας είναι χρωματισμένοι με μπλε όταν μεταφέρουν εντολή, με πράσινο όταν μεταφέρουν διεύθυνση, με μωβ όταν μεταφέρουν σήμα ελέγχου ή τμήμα της λέξης εντολής, και μαύρο όταν μεταφέρουν γενικά δεδομένα. Με πορτοκαλί σημειώνεται η γραμμή εξωτερικής διακοπής. Στο διάγραμμα δεν σημειώνονται τα σήματα ελέγχου της ΜΕ. Ακόμα, υποθέτουμε ότι η προέκταση προσήμου, όπου χρειάζεται, γίνεται στη ΜΕ.

Μετά την ανάλυση των διαφορετικών δρόμων ροής πληροφορίας στην αρχιτεκτονική μας, θα προσπαθήσουμε να δούμε για κάθε εντολή και κάθε μέθοδο διευθυνσιοδότησης ποιες μικρολειτουργίες τελούνται. Οι μικρολειτουργίες αυτές θα σχετίζονται είτε με τη λειτουργία της ΑΛΜ, είτε με τη λειτουργία της ΜΔΜ, είτε με μεταφορά δεδομένων μεταξύ υπομονάδων και καταχωρητών.

Τα σήματα ελέγχου που χρειαζόμαστε για την ενεργοποίηση αυτών είναι αντίστοιχα σήματα ελέγχου της ΑΛΜ, σήματα ελέγχου της ΜΔΜ, και σήματα επιλογής εισόδων και εξόδων (εγγραφής αντίστοιχων καταχωρητών) των υπομονάδων της ΜΕΔ. Τέτοια σήματα είναι για παράδειγμα τα σήματα επιλογής εισόδου διευθύνσεων της ΜΔΜ “Δ/Ε” (για επιλογή ανάγνωσης εντολών – διεύθυνση που λαμβάνεται από τον PC, ή δεδομένων – διεύθυνση όπως καθορίζεται από το επόμενο σήμα) και “διευθ” (για επιλογή διεύθυνσης δεδομένων είτε από τη συνένωση PC³¹⁻²⁵ || IR²⁴⁻⁰ είτε από τον DR), τα σήματα επιλογής εισόδων της ΑΛΜ “ΑΛΜ1” και “ΑΛΜ2”, τα σήματα προσπέλασης μνήμης “ανάγνωση” και “εγγραφή”, καθώς και άλλα σήματα με συμβολικά ονόματα που λίγο-πολύ δείχνουν ποιες μικρολειτουργίες ενεργοποιούν².

Σε μια ΜΕ πολλαπλών κύκλων μηχανής ανά κύκλο εντολής, κάθε φάση του κύκλου εντολής καταλαμβάνει έναν κύκλο μηχανής, αλλά δεν είναι απαραίτητο όλες οι εντολές να περνάνε από όλες τις φάσεις. Ομαδοποιώντας τα σήματα ελέγχου που χρειάζεται κάθε φάση για κάθε εντολή και αντίστοιχη διευθυνσιοδότηση, σχηματίζουμε τον πίνακα της επόμενης σελίδας, όπου κάθε γραμμή αντιστοιχεί σε μια φάση του κύκλου της αντίστοιχης εντολής, κι επομένως σε έναν κύκλο μηχανής. Η διαδοχή των φάσεων για κάθε εντολή δίνεται με την αρίθμηση της τρίτης στήλης, ξεκινώντας με τις φάσεις ανάκλησης και αποκωδικοποίησης, που είναι αριθμημένες με 0 και 1 αντίστοιχα, και είναι κοινές για όλες τις εντολές.

Στον πίνακα αυτόν παρατηρούμε ότι στη φάση αποκωδικοποίησης δεν τελείται καμία μικρολειτουργία. Για να κερδίσουμε χρόνο στον κύκλο εντολής, μπορούμε να αναθέσουμε στη φάση αποκωδικοποίησης μικρολειτουργίες από κάποια άλλη φάση που ακολουθεί.

Μελετώντας τις εντολές και διευθυνσιοδοτήσεις που έχουμε, διαπιστώνουμε ότι η φάση επίλυσης έμμεσης διευθυνσιοδότησης για όλες τις εντολές, όπως επίσης και η φάση ανάγνωσης της κατ’ ευθείαν διευθυνσιοδότησης για τις εντολές LOAD, ADD και SUB, τελούν τις ίδιες μικρολειτουργίες. Η μόνη πληροφορία που χρησιμοποιείται στις φάσεις αυτές είναι το περιεχόμενο των PC και IR, που είναι ήδη γνωστό από τη φάση ανάκλησης. Επομένως οι αντίστοιχες μικρολειτουργίες μπορούν να ανατεθούν στη φάση αποκωδικοποίησης. Με βάση την παρατήρηση αυτή ο πίνακας μικρολειτουργιών και σημάτων ελέγχου διαμορφώνεται όπως φαίνεται στη μεθεπόμενη σελίδα.

Πρέπει πάντως να σημειώσουμε ότι η ανάγνωση από τη μνήμη από τη διεύθυνση PC||IR δεν είναι πάντοτε έγκυρη μικρολειτουργία. Για παράδειγμα, οι εντολές διακλάδωσης δεν προσπελαίνουν τη μνήμη, κι επομένως, σε τέτοια εντολή, δε μπορούμε να είμαστε σίγουροι ότι η συνένωση των PC και IR θα μας δώσει έγκυρη διεύθυνση μνήμης για ανάγνωση. Για να μην έχουμε εσφαλμένη διάγνωση λάθους στην προσπέλαση της μνήμης, θα πρέπει να αποτρέψουμε τον επεξεργαστή να οδηγηθεί σε τέτοια διάγνωση, πριν την ολοκλήρωση της αποκωδικοποίησης της εντολής, οπότε θα ξέρουμε αν η εντολή οδηγεί σε προσπέλαση μνήμης ή όχι.

Κατά τα άλλα, στον πίνακα μικρολειτουργιών και σημάτων ελέγχου παρατηρούμε ακόμα:

² Μπορεί εύκολα να διαπιστωθεί η αντιστοιχία των σημάτων αυτών προς τα σήματα της ΜΕΔ MIPS που έχουμε μελετήσει. Για παράδειγμα, το σήμα “Δ/Ε” είναι το γνωστό μας “IorD”, το σήμα “ΑΛΜ1” είναι το γνωστό μας “ALUSrc1”, το σήμα “διευθ” είναι παραπλήσιο με το σήμα “Addr” που εισάγαμε στη ΜΕΔ MIPS σε προηγούμενη άσκηση, κοκ.

Εντολή	Διευθυνσιοδότηση	Φάση	Μικρολειτουργίες/Σήματα Ελέγχου
Φάση Ανάκλησης		0	$\Delta/E = 0$, ανάγνωση, εγγραφή IR, ΑΛΜ1 = PC, ΑΛΜ2 = 1, πρόσθεση, PC = ΑΛΜ, εγγραφή PC
Φάση Αποκωδικοποίησης		1	
LOAD	κατ' ευθείαν	2	$\Delta/E = 1$, ανάγνωση, διευθ = PC IR, εγγραφή DR
		3	$\Sigma = DR$, εγγραφή Σ
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = PC IR, εγγραφή DR
		3	$\Delta/E = 1$, ανάγνωση, διευθ = DR, εγγραφή DR
		4	$\Sigma = DR$, εγγραφή Σ
άμεση	2	$\Sigma = \text{προεκτ(IR)}$, εγγραφή Σ	
STORE	κατ' ευθείαν	2	$\Delta/E = 1$, εγγραφή, διευθ = PC IR
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = PC IR, εγγραφή DR
		3	$\Delta/E = 1$, εγγραφή, διευθ = DR
ADD	κατ' ευθείαν	2	$\Delta/E = 1$, ανάγνωση, διευθ = PC IR, εγγραφή DR
		3	ΑΛΜ1 = Σ , ΑΛΜ2 = DR, πρόσθεση, $\Sigma = \text{ΑΛΜ}$, εγγραφή Σ
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = PC IR, εγγραφή DR
		3	$\Delta/E = 1$, ανάγνωση, διευθ = DR, εγγραφή DR
		4	ΑΛΜ1 = Σ , ΑΛΜ2 = DR, πρόσθεση, $\Sigma = \text{ΑΛΜ}$, εγγραφή Σ
άμεση	2	ΑΛΜ1 = Σ , ΑΛΜ2 = προεκτ(IR), πρόσθεση, $\Sigma = \text{ΑΛΜ}$, εγγραφή Σ	
SUB	κατ' ευθείαν	2	$\Delta/E = 1$, ανάγνωση, διευθ = PC IR, εγγραφή DR
		3	ΑΛΜ1 = Σ , ΑΛΜ2 = DR, αφαίρεση, $\Sigma = \text{ΑΛΜ}$, εγγραφή Σ
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = PC IR, εγγραφή DR
		3	$\Delta/E = 1$, ανάγνωση, διευθ = DR, εγγραφή DR
		4	ΑΛΜ1 = Σ , ΑΛΜ2 = DR, αφαίρεση, $\Sigma = \text{ΑΛΜ}$, εγγραφή Σ
JMP	κατ' ευθείαν	2	PC = PC IR, εγγραφή PC
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = PC IR, εγγραφή DR
		3	PC = DR, εγγραφή PC
JAL	κατ' ευθείαν	2	PC = PC IR, εγγραφή PC, $\Sigma = \text{PC}$, εγγραφή Σ
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = PC IR, εγγραφή DR
		3	PC = DR, εγγραφή PC, $\Sigma = \text{PC}$, εγγραφή Σ
BNZ	σχετική	2	ΑΛΜ1 = PC, ΑΛΜ2 = προεκτ(IR), πρόσθεση, PC = ΑΛΜ, if ($\Sigma \neq 0$) εγγραφή PC
BLTZ	σχετική	2	ΑΛΜ1 = PC, ΑΛΜ2 = προεκτ(IR), πρόσθεση, PC = ΑΛΜ, if ($\Sigma < 0$) εγγραφή PC

- Οι εντολές εκτελούν λιγότερες από 5 φάσεις στον κύκλο εντολής, επειδή η φάση εύρεσης τελικής διεύθυνσης τελούμενου ενσωματώθηκε στη φάση αποκωδικοποίησης.
- Κάθε σήμα επιλογής εισόδου κάποιας υπομονάδας ή καταχωρητή που δε χρησιμοποιείται σε κάποιον κύκλο μηχανής γίνεται *αδιάφορη συνθήκη* σε εκείνον τον κύκλο. Κάθε άλλο σήμα που δε χρησιμοποιείται σε κάποιον κύκλο μηχανής πρέπει να είναι *απενεργοποιημένο* σε εκείνον τον κύκλο.
- Το σήμα “ Δ/E ” έχει τιμή 1 όταν η ΜΔΜ προσπελαύνει λέξη δεδομένων και τιμή 0 όταν η ΜΔΜ προσπελαύνει λέξη εντολής. Όταν “ Δ/E ” = 0, η διεύθυνση προέρχεται από το μετρητή προγράμματος, διαφορετικά η διεύθυνση καθορίζεται με το σήμα “διευθ”, όπως αναφέραμε νωρίτερα. Στην εγγραφή δε χρειάζεται σήμα επιλογής εισόδου δεδομένων, αφού τα δεδομένα μπορούν να έρθουν μόνο από το συσσωρευτή Σ .
- Η ΑΛΜ ελέγχεται μέσω του σήματος της πράξης “πρόσθεση” και “αφαίρεση”.
- Τόσο οι εντολές πράξεων ΑΛΜ, όσο και οι εντολές άλματος με συνθήκη, διαφοροποιούνται μεταξύ τους με το σήμα πράξης ή συνθήκης που στέλνει η ΜΕ στην ΑΛΜ, κι επομένως η διαδοχή φάσεων σ’ αυτές τις εντολές είναι ταυτόσημη. Εξαιρεση αποτελεί η εντολή ADD με άμεσο τελούμενο, εφ’ όσον δεν υποστηρίζεται αντίστοιχη διευθυνσιοδότηση στην SUB.

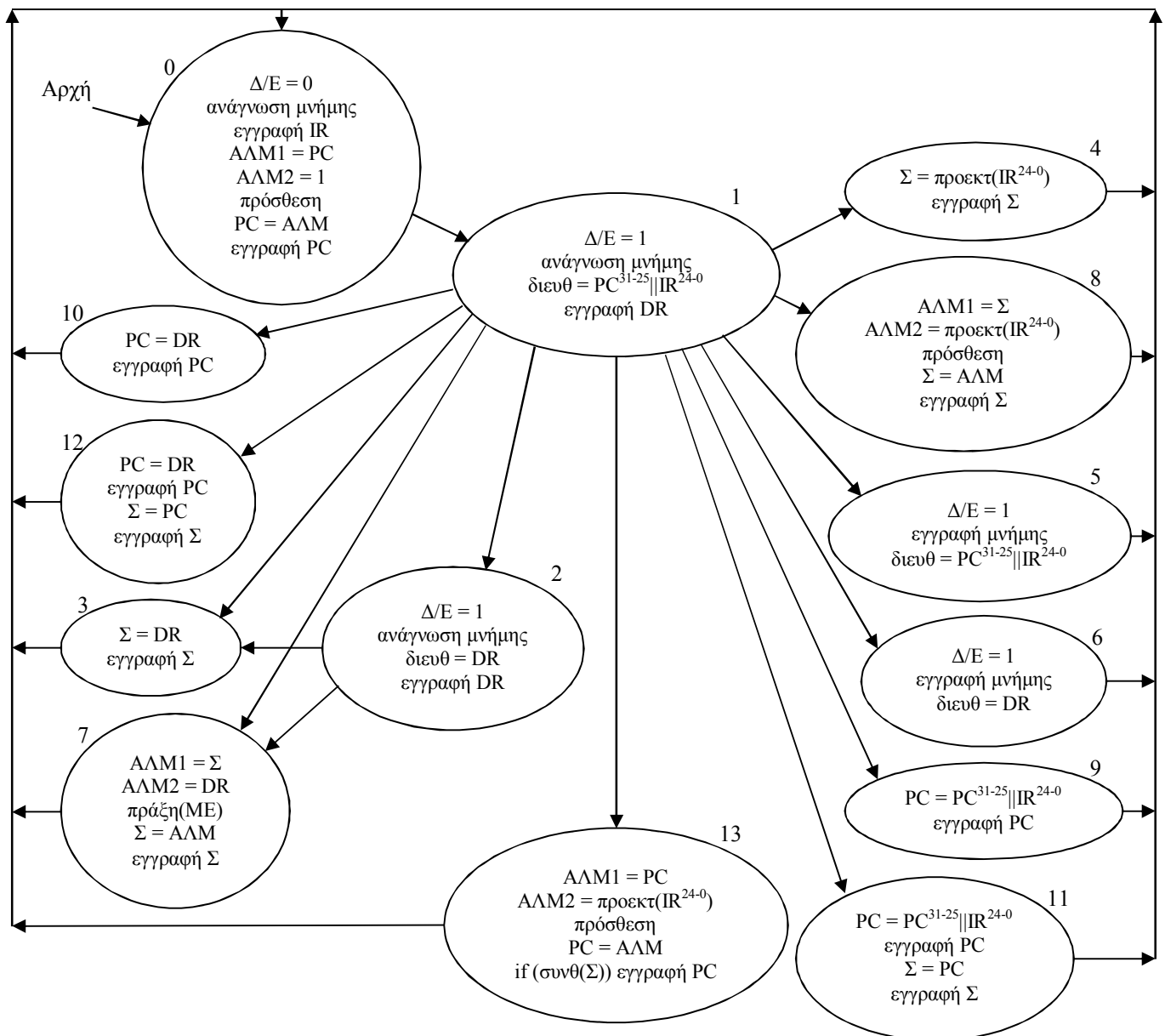
Εντολή	Διευθυνσιοδότηση	Φάση	Μικρολειτουργίες/Σήματα Ελέγχου
Φάση Ανάκλησης		0	$\Delta/E = 0$, ανάγνωση, εγγραφή IR, $ΑΛΜ1 = PC$, $ΑΛΜ2 = 1$, πρόσθεση, $PC = ΑΛΜ$, εγγραφή PC
Φάση Αποκωδικοποίησης		1	$\Delta/E = 1$, ανάγνωση, διευθ = PC IR, εγγραφή DR
LOAD	κατ' ευθείαν	2	$\Sigma = DR$, εγγραφή Σ
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = DR, εγγραφή DR
		3	$\Sigma = DR$, εγγραφή Σ
	άμεση	2	$\Sigma = \text{προεκτ}(IR)$, εγγραφή Σ
STORE	κατ' ευθείαν	2	$\Delta/E = 1$, εγγραφή, διευθ = PC IR
	έμμεση	2	$\Delta/E = 1$, εγγραφή, διευθ = DR
ADD	κατ' ευθείαν	2	$ΑΛΜ1 = \Sigma$, $ΑΛΜ2 = DR$, πρόσθεση, $\Sigma = ΑΛΜ$, εγγραφή Σ
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = DR, εγγραφή DR
		3	$ΑΛΜ1 = \Sigma$, $ΑΛΜ2 = DR$, πρόσθεση, $\Sigma = ΑΛΜ$, εγγραφή Σ
	άμεση	2	$ΑΛΜ1 = \Sigma$, $ΑΛΜ2 = \text{προεκτ}(IR)$, πρόσθεση, $\Sigma = ΑΛΜ$, εγγραφή Σ
SUB	κατ' ευθείαν	2	$ΑΛΜ1 = \Sigma$, $ΑΛΜ2 = DR$, αφαίρεση, $\Sigma = ΑΛΜ$, εγγραφή Σ
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = DR, εγγραφή DR
		3	$ΑΛΜ1 = \Sigma$, $ΑΛΜ2 = DR$, αφαίρεση, $\Sigma = ΑΛΜ$, εγγραφή Σ
JMP	κατ' ευθείαν	2	$PC = PC IR$, εγγραφή PC
	έμμεση	2	$PC = DR$, εγγραφή PC
JAL	κατ' ευθείαν	2	$PC = PC IR$, εγγραφή PC, $\Sigma = PC$, εγγραφή Σ
	έμμεση	2	$PC = DR$, εγγραφή PC, $\Sigma = PC$, εγγραφή Σ
BNZ	σχετική	2	$ΑΛΜ1 = PC$, $ΑΛΜ2 = \text{προεκτ}(IR)$, πρόσθεση, $PC = ΑΛΜ$, if ($\Sigma \neq 0$) εγγραφή PC
BLTZ	σχετική	2	$ΑΛΜ1 = PC$, $ΑΛΜ2 = \text{προεκτ}(IR)$, πρόσθεση, $PC = ΑΛΜ$, if ($\Sigma < 0$) εγγραφή PC

- Η εγγραφή κάθε καταχωρητή ελέγχεται μέσω κατάλληλου σήματος επίτρεψης. Κάτι τέτοιο δεν είναι απαραίτητο για τον DR, εφ' όσον σε καμία περίπτωση το περιεχόμενό του δε χρειάζεται να διατηρηθεί για περισσότερο από έναν κύκλο μηχανής.
- Στις εντολές άλματος με συνθήκη ο έλεγχος της συνθήκης γίνεται πάνω στην παρούσα τιμή του συσσωρευτή. Γι' αυτό και όλες οι μικρολειτουργίες μπορούν να τελεστούν στον ίδιο κύκλο μηχανής.

Στη συνέχεια θα αντιστοιχήσουμε κάθε γραμμή του παραπάνω πίνακα με κάποια κατάσταση της ME. Εκτός από τις φάσεις ανάκλησης και αποκωδικοποίησης που ως κοινές για όλες τις εντολές αντιστοιχούνται σε κοινές καταστάσεις, υπάρχουν και άλλες φάσεις που είναι κοινές για πολλές εντολές, κι επομένως μπορούν επίσης να αντιστοιχηθούν σε κοινές καταστάσεις. Έτσι, με αρχική κατάσταση την κατάσταση 0, η οποία ενεργοποιεί τις μικρολειτουργίες της φάσης ανάκλησης, η επόμενη κατάσταση 1 ενεργοποιεί τις μικρολειτουργίες της φάσης αποκωδικοποίησης. Η αποκωδικοποίηση της εντολής, λαμβάνοντας υπ' όψη τις κοινές φάσεις, μας οδηγεί σε μία από 12 καταστάσεις. Πιο συγκεκριμένα, η τελική προσπέλαση ανάγνωσης από τη μνήμη μετά από έμμεση αναφορά αποτελεί κοινή φάση για όλες τις σχετικές εντολές. Ακόμα, τόσο η κατ' ευθείαν, όσο και η έμμεση διευθυνσιοδότηση στις εντολές ADD και SUB οδηγούν σε κοινή φάση εκτέλεσης γι' αυτές. Τέλος, οι δύο εντολές άλματος με συνθήκη έχουν επίσης κοινή φάση εκτέλεσης.

Συνεχίζοντας την ανάλυση καταστάσεων για κάθε εντολή και μέθοδο διευθυνσιοδότησης που αυτή υποστηρίζει, βρίσκουμε όσες κοινές φάσεις ακολουθούν τις προηγούμενες, και καταλήγουμε στο διάγραμμα μηχανής καταστάσεων της επόμενης σελίδας.

Για να κατανοήσουμε τη μηχανή καταστάσεων, θα συμπληρώσουμε το παραπάνω διάγραμμα με τον αντίστοιχο πίνακα μετάβασης καταστάσεων. Υποθέτουμε ότι ο κωδικός λειτουργίας Op κάθε εντολής παίρνει τιμή από 0 μέχρι 7, με τη σειρά που αυτές μας δίνονται στην εκφώνηση. Ακόμα, υποθέτουμε ότι τα ψηφία διευθυνσιοδότησης $\Delta \equiv IR^{28-25}$ αντιστοιχούν στις μεθόδους διευθυνσιοδότησης, επίσης με τη σειρά που αυτές μας δίνονται στην εκφώνηση, με το Δ_0 να υποδηλώνει κατ' ευθείαν διευθυνσιοδότηση, το Δ_1 να υποδηλώνει άμεση διευθυνσιο-



δότηση, κοκ. Σε κάθε περίπτωση, μόνο ένα ψηφίο διευθυνσιοδότησης επιτρέπεται να έχει τιμή 1.

Για να βρούμε τη συνθήκη μετάβασης $C(i,j)$ από μια κατάσταση i σε μια κατάσταση j , όταν υπάρχουν επιλογές μεταβάσεων, πρέπει να εξετάσουμε τι εντολή εκτελείται και με ποια μέθοδο διευθυνσιοδότησης. Αν δεν υπάρχουν επιλογές μεταβάσεων από την κατάσταση i , θεωρούμε ότι η μοναδική μετάβαση γίνεται με συνθήκη 1. Έτσι, σχηματίζουμε τον ακόλουθο πίνακα:

Συνθήκη Μετάβασης	Τιμή
$C(1,2)$	$\Delta_2 * ((Op==LOAD) + (Op==ADD) + (Op==SUB))$
$C(1,3)$	$\Delta_0 * (Op==LOAD)$
$C(1,4)$	$\Delta_1 * (Op==LOAD)$
$C(1,5)$	$\Delta_0 * (Op==STORE)$
$C(1,6)$	$\Delta_2 * (Op==STORE)$
$C(1,7)$	$\Delta_0 * ((Op==ADD) + (Op==SUB))$
$C(1,8)$	$\Delta_1 * (Op==ADD)$
$C(1,9)$	$\Delta_0 * (Op==JMP)$

C(1,10)	$\Delta_2 * (Op==JMP)$
C(1,11)	$\Delta_0 * (Op==JAL)$
C(1,12)	$\Delta_2 * (Op==JAL)$
C(1,13)	$\Delta_3 * ((Op==BNZ) + (Op==BLTZ))$
C(2,3)	$(Op==LOAD)$
C(2,7)	$((Op==ADD) + (Op==SUB))$

όπου για απλούστευση δεν αναγράψαμε τις μεταβάσεις που γίνονται με συνθήκη 1.

Για παράδειγμα, η μετάβαση από την κατάσταση 1 στην κατάσταση 2 γίνεται όταν έχουμε έμμεση διευθυνσιοδότηση σε μια από τις εντολές LOAD, ADD και SUB. Όταν έχουμε έμμεση διευθυνσιοδότηση σε εντολή STORE, μεταβαίνουμε από την κατάσταση 1 στην κατάσταση 6, στην οποία γίνεται η αποθήκευση στη σωστή διεύθυνση μνήμης που λαμβάνεται από τον DR. Όταν έχουμε έμμεση διευθυνσιοδότηση σε εντολή άλματος, η μετάβαση από την κατάσταση 1 γίνεται προς μία από τις 10 και 12.

Άμεση διευθυνσιοδότηση μπορούμε να έχουμε μόνο για εντολή LOAD, οπότε από την κατάσταση 1 μεταβαίνουμε στην 4, ή για εντολή ADD, οπότε από την κατάσταση 1 μεταβαίνουμε στην 8.

Από την κατάσταση 2 μεταβαίνουμε στην κατάσταση 3 μόνο όταν έχουμε έμμεση διευθυνσιοδότηση σε εντολή LOAD. Όταν βρισκόμαστε στην κατάσταση 2, έχουμε ήδη εξασφαλίσει την έμμεση διευθυνσιοδότηση, κι επομένως για τη συνθήκη μετάβασης αρκεί ο έλεγχος του κωδικού λειτουργίας της εντολής.

Στη συνέχεια θα προχωρήσουμε στην τελική φάση της σχεδίασης της Μονάδας Ελέγχου της αρχιτεκτονικής μας. Αυτό που χρειαζόμαστε είναι να βρούμε τις λογικές συναρτήσεις που εκφράζουν τόσο τις επόμενες καταστάσεις, όσο και τα σήματα ελέγχου, σαν συνδυασμό των τιμών των εισόδων και των παρουσών καταστάσεων. Οι εισοδοί στην περίπτωσή μας είναι ο κωδικός λειτουργίας και τα ψηφία διευθυνσιοδότησης της λέξης εντολής.

Όσο αφορά τις επόμενες καταστάσεις, ο παραπάνω πίνακας μετάβασης καταστάσεων αρκεί για τον υπολογισμό των αντίστοιχων συναρτήσεων. Αν κωδικοποιήσουμε τη λέξη κατάστασης S με την τεχνική του ενός ενεργού σήματος, σε κάθε κατάσταση θα αντιστοιχεί ένα bit της λέξης αυτής. Εναλλακτικά, μπορούμε να κωδικοποιήσουμε τη λέξη κατάστασης με 4 bits, εφ' όσον έχουμε 14 καταστάσεις, οπότε η λέξη κατάστασης θα περιέχει τη δυαδική αναπαράσταση του αριθμού κατάστασης. Σε κάθε περίπτωση, πρέπει για κάθε bit της λέξης κατάστασης να υπολογίσουμε την αντίστοιχη λογική συνάρτηση.

Έστω ότι χρησιμοποιούμε την τεχνική ενός ενεργού σήματος για την κωδικοποίηση της λέξης κατάστασης. Από τον παραπάνω πίνακα, αναπτύσσοντας τον κωδικό λειτουργίας στα 3 επιμέρους ψηφία του Op2, Op1 και Op0, θα πάρουμε:

$$\begin{aligned}
 S_2 &= S_1 \Delta_2 (Op_2' Op_1' Op_0' + Op_2' Op_1 Op_0' + Op_2' Op_1 Op_0) = \\
 &= S_1 \Delta_2 Op_2' (Op_0' + Op_1) \\
 S_3 &= S_1 \Delta_0 Op_2' Op_1' Op_0' + S_2 Op_2' Op_1' Op_0' = \\
 &= (S_1 \Delta_0 + S_2) Op_2' Op_1' Op_0' \\
 S_4 &= S_1 \Delta_1 Op_2' Op_1' Op_0' \\
 S_5 &= S_1 \Delta_0 Op_2' Op_1' Op_0 \\
 S_6 &= S_1 \Delta_2 Op_2' Op_1' Op_0 \\
 S_7 &= S_1 \Delta_0 (Op_2' Op_1 Op_0' + Op_2' Op_1 Op_0) + S_2 (Op_2' Op_1 Op_0' + Op_2' Op_1 Op_0) = \\
 &= (S_1 \Delta_0 + S_2) Op_2' Op_1 \\
 S_8 &= S_1 \Delta_1 Op_2' Op_1 Op_0' \\
 S_9 &= S_1 \Delta_0 Op_2 Op_1' Op_0' \\
 S_{10} &= S_1 \Delta_2 Op_2 Op_1' Op_0' \\
 S_{11} &= S_1 \Delta_0 Op_2 Op_1' Op_0 \\
 S_{12} &= S_1 \Delta_2 Op_2 Op_1' Op_0 \\
 S_{13} &= S_1 \Delta_3 (Op_2 Op_1 Op_0' + Op_2 Op_1 Op_0) = \\
 &= S_1 \Delta_3 Op_2 Op_1
 \end{aligned}$$

και για τις μεταβάσεις με συνθήκη 1:

$$\begin{aligned} S_1 &= S_0 \\ S_0 &= S_3 + S_4 + S_5 + S_6 + S_7 + S_8 + S_9 + S_{10} + S_{11} + S_{12} + S_{13} \end{aligned}$$

Όσο αφορά τα σήματα ελέγχου της αρχιτεκτονικής, οι αντίστοιχες λογικές εκφράσεις υπολογίζονται ακόμα πιο εύκολα απ' ό,τι οι παραπάνω εκφράσεις της επόμενης κατάστασης. Ένα σημαντικό χαρακτηριστικό των κυκλωμάτων ελέγχου που σχεδιάζονται με αυτή τη διαδικασία είναι πως οι έξοδοι σε κάθε κύκλο μηχανής καθορίζονται από την παρούσα κατάσταση και μόνο από αυτήν. Μ' άλλα λόγια, οι ζητούμενες εκφράσεις για όλα τα σήματα ελέγχου θα είναι το λογικό Η των σημάτων της λέξης κατάστασης που αντιστοιχούν σε καταστάσεις που ενεργοποιούν το κάθε σήμα. Σε μεμονωμένες εξαιρέσεις, στον υπολογισμό των εκφράσεων συμμετέχουν και άλλα σήματα που παίρνουν τιμή σε αυτές τις καταστάσεις.

Για παράδειγμα, το σήμα "ανάγνωση" που αφορά τη ΜΔΜ παίρνει τιμή από τις καταστάσεις που περιέχουν ανάγνωση από τη μνήμη:

$$\text{"ανάγνωση"} = S_0 + S_1 + S_2$$

ενώ το αντίστοιχο σήμα "εγγραφή" έχει τιμή:

$$\text{"εγγραφή"} = S_5 + S_6$$

και το σήμα "Δ/Ε" που προσδιορίζει αν η τιμή που προσπελαύνεται στη μνήμη είναι δεδομένο ή εντολή:

$$\text{"Δ/Ε"} = S_1 + S_2 + S_5 + S_6$$

Το σήμα "εγγραφή Σ" που αφορά το συσσωρευτή παίρνει τιμή από τις καταστάσεις που γράφουν στον Σ:

$$\text{"εγγραφή Σ"} = S_3 + S_4 + S_7 + S_8 + S_{11} + S_{12}$$

Τα σήματα εγγραφής των υπόλοιπων καταχωρητών θα είναι:

$$\text{"εγγραφή IR"} = S_0$$

$$\text{"εγγραφή DR"} = S_1 + S_2$$

$$\text{"εγγραφή PC"} = S_0 + S_9 + S_{10} + S_{11} + S_{12} + S_{13} \text{ "συνθ(Σ)"}$$

Τέλος, το σήμα "πρόσθεση" έχει τιμή:

$$\text{"πρόσθεση"} = S_0 + S_7 \text{ "πράξη(ME)"} + S_8 + S_{13}$$

Με παρόμοιο τρόπο καθορίζονται τα σήματα επιλογής εισόδων των υπομονάδων της ΜΕΔ.

Η είσοδος στον Σ μπορεί να γίνει από την ΑΛΜ, από τον PC, από τον IR (με προέκταση προσήμου) ή από τον DR. Αν ένας πολυπλέκτης 4×1 ΜΣ απαιτείται γι' αυτή την επιλογή, κι αν οι επιλογές κωδικοποιούνται με τη σειρά που αναφέρθηκαν – "00" για την πρώτη, "01" για τη δεύτερη, "10" για την τρίτη και "11" για την τέταρτη, τα ψηφία επιλογής θα είναι:

$$\text{"ΜΣ}_1\text{"} = S_3 + S_4$$

$$\text{"ΜΣ}_0\text{"} = S_3 + S_{11} + S_{12}$$

Η είσοδος διευθύνσεων της ΜΔΜ μπορεί να γίνει από τον PC, τον IR (με παράθεση ψηφίων του PC) ή τον DR. Αντί να χρησιμοποιήσουμε έναν πολυπλέκτη 4×1, μπορούμε εναλλακτικά να χρησιμοποιήσουμε δύο μικρότερους, έναν πολυπλέκτη 2×1 που επιλέγει μεταξύ του PC και της εξόδου ενός δεύτερου πολυπλέκτη 2×1, ο οποίος επιλέγει μεταξύ των υπόλοιπων καταχωρητών. Αν ο πρώτος πολυπλέκτης επιλέγει τον PC με τιμή επιλογής 0, το σήμα επιλογής του θα είναι απλά το "Δ/Ε". Όσο αφορά το δεύτερο πολυπλέκτη, αν οι επιλογές του κωδικοποιούνται με τη σειρά που αναφέραμε τους καταχωρητές, το σήμα επιλογής του θα είναι το "διευθ":

$$\text{"διευθ"} = S_2 + S_6$$

Η επιλογή εισόδων της ΑΛΜ γίνεται με δύο πολυπλέκτες. Η είσοδος ΑΛΜ1 επιλέγει μεταξύ του Σ και του PC, οπότε απαιτεί έναν πολυπλέκτη 2×1. Η είσοδος ΑΛΜ2 επιλέγει μεταξύ του DR, του IR (με προέκταση προσήμου) και της σταθεράς 1, οπότε απαιτεί έναν πολυπλέκτη 4×1. Αν οι παραπάνω επιλογές κωδικοποιούνται με τη σειρά που αναφέρθηκαν, τότε:

$$\text{"ΑΛΜ1"} = S_0 + S_{13}$$

$$\text{"ΑΛΜ2}_1\text{"} = S_0$$

$$\text{"ΑΛΜ2}_0\text{"} = S_8 + S_{13}$$

Η επιλογή εισόδων του καταχωρητή PC γίνεται με παρόμοιο τρόπο. Ο PC δέχεται είσοδο από τον IR (με παράθεση ψηφίων του PC), από τον DR και από την ΑΛΜ. Έτσι, ο έλεγχος του αντίστοιχου πολυπλέκτη 4×1 MPC, με κωδικοποίηση εισόδων σύμφωνα με τη σειρά που αυτές αναφέρθηκαν, θα γίνεται ως εξής:

$$\text{“MPC}_1\text{”} = S_0 + S_{13}$$

$$\text{“MPC}_0\text{”} = S_{10} + S_{12}$$

Εξαίρεση στον κανόνα που αναφέραμε για τη μορφή των λογικών εκφράσεων των σημάτων ελέγχου αποτελούν σήματα επιλογής μικρολειτουργιών υπομονάδων που δεν είναι διαμοιρασμένες σε διαφορετικές καταστάσεις, αλλά είναι συγκεντρωμένες σε μια γενικευμένη μικρολειτουργία. Για παράδειγμα, η έκφραση “πράξη(ME)” στην κατάσταση 7 υποδηλώνει το σύνολο των διαφορετικών πράξεων που εκτελούνται στην ΑΛΜ. Παρόμοια, η έκφραση “συνθ(Σ)” υποδηλώνει το σύνολο των διαφορετικών συνθηκών του συσσωρευτή Σ που χρησιμοποιούνται από τον PC στην κατάσταση 13. Τέτοια σήματα ελέγχου χρησιμοποιούν τη λέξη εντολής ή κάποια ψηφία κατάστασης, για να επιλέξουν την ακριβή μικρολειτουργία.

Έτσι, δεδομένου ότι στην ΑΛΜ γίνονται μόνο δύο πράξεις, το σήμα “πράξη(ME)” θα παίρνει δύο τιμές. Αν υποθέσουμε ότι η πρόσθεση κωδικοποιείται στον αριθμό 1, ώστε το σήμα “πρόσθεση” που αναφέραμε πιο πάνω να παίρνει τη σωστή τιμή, το σήμα αυτό θα είναι:

$$\text{“πράξη(ME)”} = Op_2'Op_1Op_0'$$

Η τελική επιλογή πράξης στην ΑΛΜ γίνεται όπως είπαμε με το σήμα “πρόσθεση”.

Τέλος, όσο αφορά το σήμα “συνθ(Σ)”, θα χρησιμοποιήσουμε τόσο τη λέξη εντολής, όσο και τα ψηφία ελέγχου της τιμής του Σ. Έτσι, το σήμα αυτό θα γίνεται:

$$\text{“συνθ(Σ)”} = Z'Op_2Op_1Op_0' + NOp_2Op_1Op_0$$

Άσκηση 4:

Θεωρήστε την αρχιτεκτονική συσσωρευτή της προηγούμενης άσκησης, στην οποία θέλουμε να προσθέσουμε δεικτοδοτούμενη διευθυνσιοδότηση μέσω του συσσωρευτή Σ.

Σκεφτείτε τα προβλήματα που παρουσιάζονται από τη χρήση του Σ για τη δεικτοδότηση, και προτείνετε τρόπους για την αντιμετώπισή τους. Τροποποιήστε κατάλληλα τη ΜΕΔ της αρχιτεκτονικής, και δώστε τον πίνακα μικρολειτουργιών και σημάτων ελέγχου καθώς και το νέο διάγραμμα της μηχανής καταστάσεων με τον αντίστοιχο πίνακα μετάβασης. Τέλος, ξαναδώστε τις λογικές συναρτήσεις επόμενης κατάστασης και σημάτων ελέγχου της αρχιτεκτονικής.

Απάντηση:

Για να προσθέσουμε δεικτοδότηση στην προσπέλαση μνήμης κάποιας αρχιτεκτονικής, χρειαζόμαστε έναν καταχωρητή-δείκτη, έναν καταχωρητή δηλαδή που θα περιέχει τη μετατόπιση που προστίθεται στη διεύθυνση βάσης. Στην περίπτωση μιας αρχιτεκτονικής φόρτωσης-αποθήκευσης, όπως η MIPS, ο καταχωρητής-δείκτης μπορεί να είναι ένας από τους καταχωρητές γενικού σκοπού. Στην περίπτωση μιας αρχιτεκτονικής συσσωρευτή όμως, ο καταχωρητής-δείκτης πρέπει να προστεθεί σαν επιπλέον καταχωρητής. Εναλλακτικά, και όπως ζητείται στην παρούσα άσκηση, σαν καταχωρητής-δείκτης μπορεί να χρησιμοποιηθεί ο συσσωρευτής Σ. Στην αρχιτεκτονική συσσωρευτή η διεύθυνση βάσης παρέχεται από τη λέξη εντολής.

Σε κάθε περίπτωση που θέλουμε να χρησιμοποιήσουμε δεικτοδοτούμενη διευθυνσιοδότηση, ο καταχωρητής-δείκτης πρέπει επομένως να περιέχει τη μετατόπιση. Έτσι, δε μπορεί να χρησιμοποιηθεί σαν καταχωρητής-δείκτης κάποιος καταχωρητής που περιέχει άλλο τελούμενο εισόδου, ή ισοδύναμα, η δεικτοδοτούμενη διευθυνσιοδότηση δε μπορεί να υποστηριχθεί αν ο καταχωρητής-δείκτης περιέχει κάποιο άλλο τελούμενο εισόδου της εντολής. Στην περίπτωση μας, που σαν καταχωρητή-δείκτη χρησιμοποιούμε το Σ, οι μόνες εντολές που μπορούν να δεχτούν δεικτοδότηση είναι οι LOAD, JMP και JAL, ενώ δε μπορούν να δεχτούν δεικτοδότηση οι εντολές STORE, ADD και SUB, επειδή στις εντολές αυτές ο Σ περιέχει το δεδομένο που αποθηκεύεται, προστίθεται ή μειώνεται αντίστοιχα.

Το παραπάνω είναι και το πιο σημαντικό πρόβλημα της δεικτοδότησης μέσω του Σ. Επειδή δεικτοδότηση χρησιμοποιούμε κυρίως για υλοποίηση διανυσματικών πράξεων, μειώνεται η

αποτελεσματικότητά της αν δεν υποστηρίζεται σε εντολές ΑΛΜ, και ακόμα περισσότερο αν δεν υποστηρίζεται σε εντολές STORE.

Ας δούμε για παράδειγμα την εντολή C:

$$a[i] = b[i] + c[i];$$

όπου συμμετέχουν τρία διανύσματα ακεραίων a, b και c. Η εντολή αυτή ιδανικά εκτελείται με δεικτοδοτούμενη φόρτωση του b, δεικτοδοτούμενη πρόσθεση του c και δεικτοδοτούμενη αποθήκευση στο a.

Ας δούμε ακόμα την εντολή C:

$$a[i] += x;$$

όπου συμμετέχει ένα διάνυσμα ακεραίων a και ένας βαθμωτός ακεραίος x. Η εντολή αυτή ιδανικά εκτελείται με δεικτοδοτούμενη φόρτωση του a, πρόσθεση του x και δεικτοδοτούμενη αποθήκευση στο a.

Καμία από τις πιο πάνω εντολές δε μπορεί να εκτελεστεί με δεικτοδότηση στην αρχιτεκτονική μας.

Οι δύο πιο πάνω εντολές περιγράφουν δύο χαρακτηριστικές κατηγορίες διανυσματικών πράξεων, μία στην οποία συμμετέχουν τουλάχιστον δύο διανύσματα, και μία στην οποία συμμετέχει ακριβώς ένα διάνυσμα. Έτσι, στην πρώτη εντολή, λόγω της συμμετοχής τριών διανυσμάτων, χρησιμοποιούμε τρεις τελικές διευθύνσεις διανυσματικών στοιχείων, κάτι που δε μπορεί να υποστηριχθεί στην αρχιτεκτονική μας, εφ' όσον δε διαθέτουμε ανεξάρτητο καταχωρητή-δείκτη. Γενικά λοιπόν, δε θα μπορέσουμε να υποστηρίξουμε δεικτοδότηση για την περίπτωση διανυσματικών πράξεων, όπου συμμετέχουν περισσότερα από ένα διάνυσμα. Στη δεύτερη εντολή όμως, χρησιμοποιούμε μόνο μία τελική διεύθυνση διανυσματικού στοιχείου, η οποία υπολογίζεται για κάποια εντολή φόρτωσης. Εάν βρούμε έναν τρόπο να διατηρήσουμε την τελική διεύθυνση μέχρι την αντίστοιχη εντολή αποθήκευσης, θα μπορέσουμε να υποστηρίξουμε δεικτοδότηση για την εντολή αυτή, και να υλοποιήσουμε τις στοιχειώδεις διανυσματικές πράξεις της δεύτερης κατηγορίας. Θα δούμε στη συνέχεια πώς μπορούμε να λύσουμε αυτό το πρόβλημα.

Γενικά, για την υποστήριξη οποιασδήποτε μεθόδου διευθυνσιοδότησης που απαιτεί κάποια βήματα για την εύρεση της τελικής διεύθυνσης προσπέλασης, είναι απαραίτητο να υπάρχουν ένας ή και περισσότεροι καταχωρητές, οι οποίοι να αποθηκεύουν προσωρινά το αποτέλεσμα κάθε βήματος. Ας θυμηθούμε ότι στην έμμεση διευθυνσιοδότηση που είδαμε στην προηγούμενη άσκηση, όπου η τελική διεύθυνση προσπέλασης παρέχεται από τη μνήμη, χρησιμοποιήσαμε τον DR, για να αποθηκεύσουμε προσωρινά την τελική διεύθυνση προσπέλασης. Κάτι τέτοιο ήταν αυτονόητο, καθώς η ανάγνωση από τη μνήμη φέρνει ούτως ή άλλως τη διεύθυνση στον DR. Στη δεικτοδοτούμενη διευθυνσιοδότηση όμως, η τελική διεύθυνση προσπέλασης υπολογίζεται στην ΑΛΜ, κι επομένως είναι απαραίτητο η έξοδος της ΑΛΜ να αποθηκεύεται σε κάποιο νέο καταχωρητή ειδικού σκοπού, έστω AR, ώστε να χρησιμοποιηθεί σε επόμενο κύκλο μηχανής για την προσπέλαση της μνήμης³.

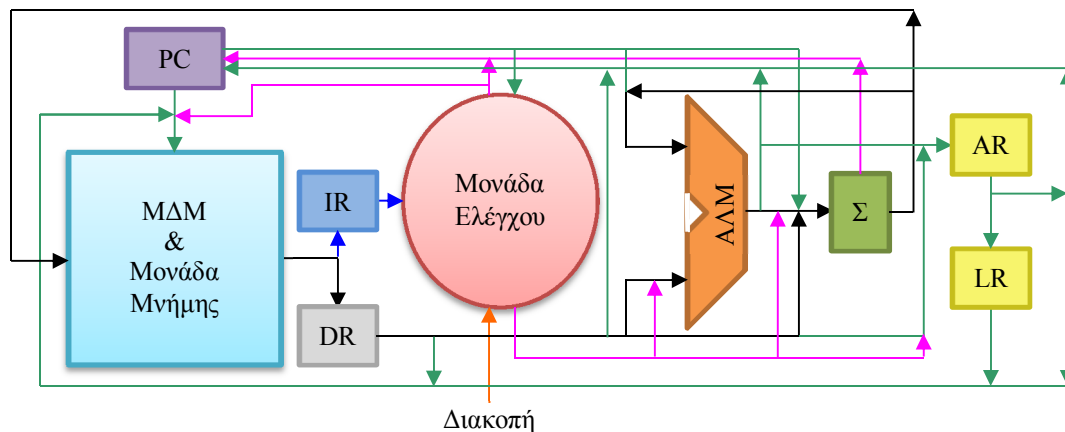
Η αντιμετώπιση του προβλήματος της χρήσης του Σ σαν καταχωρητή-δείκτη θα μπορούσε να γίνει με χρήση του AR. Για απλούστευση, ώστε να πάρουμε τελικά μικρότερο αριθμό καταστάσεων, όπως θα δούμε παρακάτω, θα εισάγουμε ένα νέο καταχωρητή ειδικού σκοπού LR, ο οποίος να αποθηκεύει την τελική διεύθυνση προσπέλασης της πιο πρόσφατης εντολής LOAD. Με τη βοήθεια του καταχωρητή αυτού, θα εισάγουμε μια υπονοούμενη διευθυνσιοδότηση, η οποία να χρησιμοποιείται από εντολές STORE, και να χρησιμοποιεί το περιεχόμενο του LR ως τελική διεύθυνση προσπέλασης.

Έτσι, καταλήγουμε σε μια αρχιτεκτονική συσσωρευτή, η οποία θα υποστηρίζει τις δύο νέες μεθόδους διευθυνσιοδότησης για τις εντολές που διαθέτουμε, τη δεικτοδοτούμενη και την υπονοούμενη, σύμφωνα με τον πιο κάτω πίνακα:

³ Αντί για το νέο καταχωρητή AR, για την προσωρινή αποθήκευση της εξόδου της ΑΛΜ θα μπορούσε να χρησιμοποιηθεί και ο DR. Ο αναγνώστης μπορεί να προχωρήσει σε λύση της άσκησης με αυτόν τον τρόπο.

Διευθυνσιοδότηση	Εντολή							
	LOAD	STORE	ADD	SUB	JMP	JAL	BNZ	BLTZ
Δεικτοδοτούμενη	1	0	0	0	1	1	0	0
Υπονοούμενη	0	1	0	0	0	0	0	0

Η ΜΕΔ της αρχιτεκτονικής αυτής θα περιγράφεται από το πιο κάτω διάγραμμα:



Για τη συνέχεια, υποθέτουμε ότι από τα 32 bits κάθε εντολής, τα 3 πιο σημαντικά περιέχουν τον κωδικό λειτουργίας, τα 6 επόμενα τη μέθοδο διευθυνσιοδότησης (ένα ψηφίο ανά μέθοδο), και τα υπόλοιπα το πιθανό τελούμενο. Αν αυτό είναι διεύθυνση, τα 9 πιο σημαντικά ψηφία του λαμβάνονται από το μετρητή προγράμματος. Διαφορετικά, προέκταση προσήμου δίνει αριθμό (δεδομένο ή μετατόπιση) μεγέθους 32 bits.

Σε αντίθεση με την προηγούμενη άσκηση, όπου οι μέθοδοι διευθυνσιοδότησης δεν ήταν δυνατό να συνδυαστούν μεταξύ τους, εδώ η δεικτοδοτούμενη διευθυνσιοδότηση μπορεί να συνδυαστεί με την έμμεση. Για απλούστευση, θα υποστηρίξουμε μόνο έμμεση δεικτοδοτούμενη διευθυνσιοδότηση, αφήνοντας την υποστήριξη της δεικτοδοτούμενης έμμεσης για άσκηση⁴.

Όπως κάναμε και στην προηγούμενη άσκηση, ας δούμε αναλυτικά ποιος είναι ο ρόλος του κάθε καταχωρητή της αρχιτεκτονικής μας:

- Ο καταχωρητής IR περιέχει τη λέξη εντολής που διαβάζεται από τη μνήμη στη φάση ανάκλησης. Το περιεχόμενό του χρησιμοποιείται από τη ΜΕ και προωθείται σε άλλες υπομονάδες της ΜΕΔ. Πιο συγκεκριμένα, η παράθεση των 9 πιο σημαντικών ψηφίων του PC με τα 23 λιγότερο σημαντικά ψηφία του IR ($PC^{31-23} || IR^{22-0}$) προωθούνται στην είσοδο διευθύνσεων της ΜΑΜ, στον PC, στην ΑΛΜ, αλλά και στον AR, για λόγο που θα εξηγήσουμε πιο κάτω. Η προέκταση προσήμου των 23 λιγότερο σημαντικών ψηφίων του IR προωθούνται στην ΑΛΜ και στον Σ.
- Ο καταχωρητής DR περιέχει τη λέξη δεδομένων που διαβάζεται από τη μνήμη, τόσο στη φάση τελικής προσπέλασης στη μνήμη, όσο και στη φάση εύρεσης τελικής διεύθυνσης τελούμενου, όταν έχουμε έμμεση διευθυνσιοδότηση. Το περιεχόμενο του DR προωθείται στην ΑΛΜ, στην είσοδο διευθύνσεων της ΜΑΜ, στον PC, στον Σ, καθώς και στον AR.
- Ο μετρητής προγράμματος PC δέχεται κάποια διεύθυνση τόσο από τους καταχωρητές IR, DR και AR, όσο και από την ΑΛΜ, απ' όπου δέχεται είτε την αυξημένη κατά 1 τιμή του είτε τη διεύθυνση προορισμού άλματος για εντολές διακλάδωσης. Ο PC δέχεται επίσης κάποιο σήμα ελέγχου από τον Σ, που μεταφέρει την τιμή της συνθήκης εντολών διακλά-

⁴ Στην έμμεση δεικτοδοτούμενη διευθυνσιοδότηση, εφ' όσον η λέξη έμμεση προσδιορίζει όλη την έκφραση δεικτοδοτούμενη διευθυνσιοδότηση, εννοείται ότι πρώτα εφαρμόζεται η δεικτοδοτούμενη και ύστερα η έμμεση διευθυνσιοδότηση για τον υπολογισμό της τελικής διεύθυνσης του τελούμενου. Το αντίστροφο θα ισχύει για τη δεικτοδοτούμενη έμμεση διευθυνσιοδότηση.

δωσης. Το περιεχόμενο του PC προωθείται στην είσοδο διευθύνσεων της ΜΔΜ, στην ΑΔΜ, στον Σ, αλλά και στη ΜΕ.

- Ο συσσωρευτής Σ που είναι ο μόνος καταχωρητής άμεσα προσπελάσιμος από κάποιο πρόγραμμα σε γλώσσα μηχανής της αρχιτεκτονικής, δέχεται τιμές από την ΑΔΜ και από όλους τους υπόλοιπους καταχωρητές, εκτός των AR και LR. Το περιεχόμενό του το στέλνει στην ΑΔΜ και στη ΜΔΜ για αποθήκευση σε εντολή STORE. Στέλνει δε και στον PC την πληροφορία για μηδενική ή αρνητική τιμή όπως ήδη προαναφέραμε, χρησιμοποιώντας για το σκοπό αυτό τα δύο ψηφία ελέγχου της τιμής του, το Z για μηδενική και το N για αρνητική τιμή.
- Ο καταχωρητής AR αποθηκεύει την ενδιάμεση διεύθυνση κάποιου τελούμενου σε περίπτωση που έχουμε δεικτοδοτούμενη διευθυνσιοδότηση. Η εύρεση αυτής της διεύθυνσης γίνεται με πρόσθεση στην ΑΔΜ της τιμής του Σ με την τιμή $PC^{31-23} \parallel IR^{22-0}$ που παραλαμβάνεται από τη ΜΕ. Ο AR στέλνει το περιεχόμενό του στην είσοδο διευθύνσεων της ΜΔΜ, στον PC για υποστήριξη δεικτοδοτούμενης εντολής άλματος και στον LR.
- Ο καταχωρητής LR είναι απαραίτητος για την υλοποίηση υπονοούμενης διευθυνσιοδότησης. Σ' αυτόν γράφεται η τελική διεύθυνση ανάγνωσης από τη μνήμη σε εντολές LOAD. Η διεύθυνση αυτή μπορεί να προέλθει από οποιονδήποτε καταχωρητή ειδικού σκοπού που χρησιμοποιήθηκε για τον υπολογισμό αυτής της διεύθυνσης, και ο οποίος θα είναι ένας από τους IR, DR και AR. Όπως θα δούμε αργότερα, μας είναι πιο εύκολο να αποθηκεύσουμε αυτή τη διεύθυνση στον AR και από αυτόν να την προωθήσουμε στον LR. Το περιεχόμενο του LR στέλνεται μόνο στην είσοδο διευθύνσεων της ΜΔΜ.

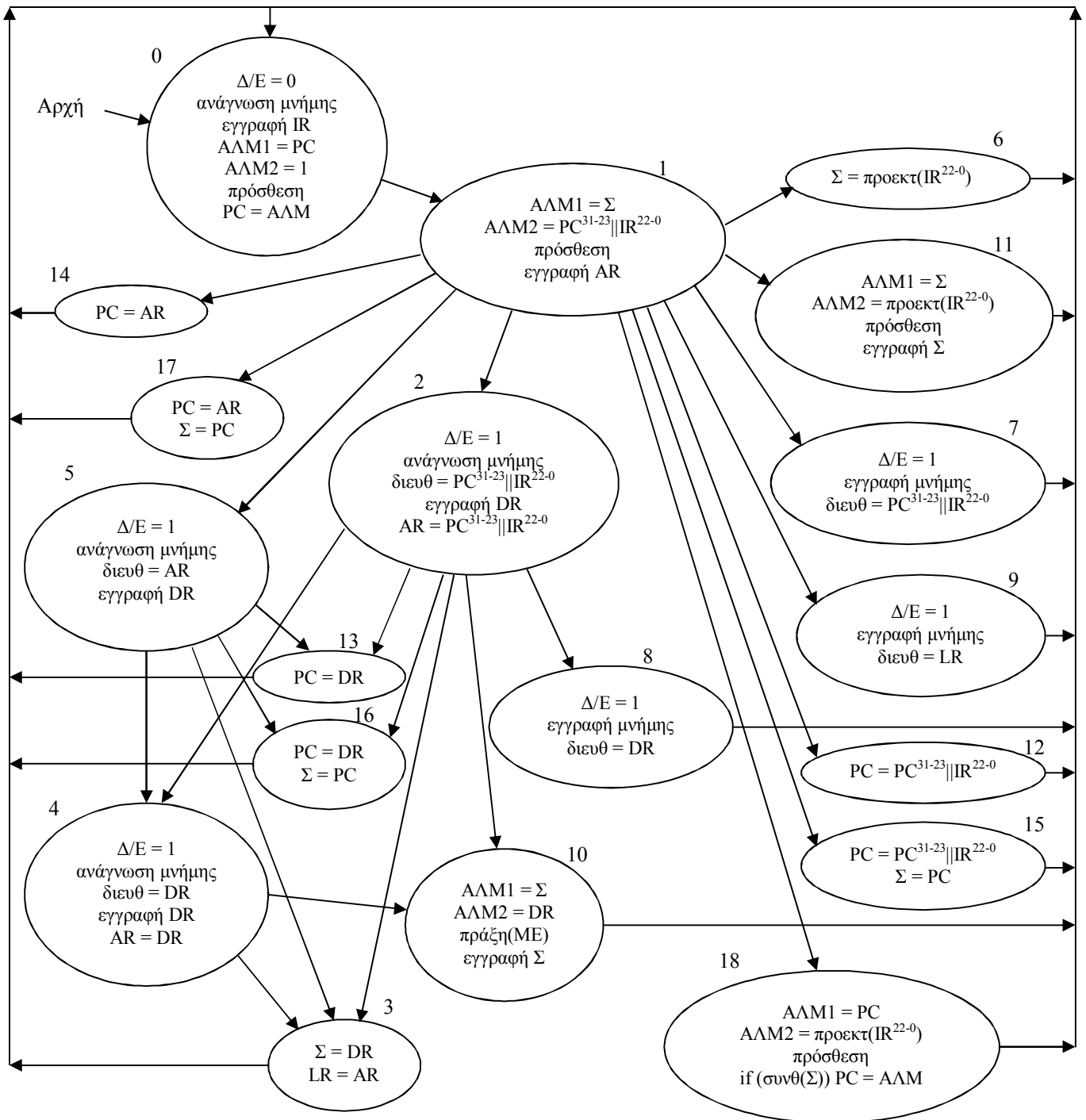
Ας προσπαθήσουμε να δούμε τώρα, για κάθε εντολή και κάθε μέθοδο διευθυνσιοδότησης, ποιες μικρολειτουργίες τελούνται. Ομαδοποιώντας τα σήματα ελέγχου που ενεργοποιούν τις μικρολειτουργίες για κάθε φάση καθεμιάς εντολής και αντίστοιχης μεθόδου διευθυνσιοδότησης, σχηματίζουμε τον πίνακα της επόμενης σελίδας. Για τον πίνακα αυτόν:

- Οι φάσεις ανάκλησης και αποκωδικοποίησης είναι κοινές για όλες τις εντολές.
- Η εύρεση τελικής διεύθυνσης τελούμενου περιλαμβάνει δύο προαιρετικές φάσεις, μία για την επίλυση της δεικτοδοτούμενης, και μία για την επίλυση της έμμεσης διευθυνσιοδότησης, που μπορεί να είναι και οι δύο παρούσες.
- Όπως είδαμε και στην προηγούμενη άσκηση, η φάση αποκωδικοποίησης μπορεί να εκτελεί μικρολειτουργίες από κάποια άλλη φάση, ώστε να μειωθεί ο συνολικός χρόνος του κύκλου εντολής. Επιλέξαμε εδώ τη φάση επίλυσης δεικτοδοτούμενης διευθυνσιοδότησης, επειδή προηγείται κάθε άλλης φάσης όταν αυτή υπάρχει, επειδή είναι ακίνδυνη (δεν οδηγεί σε λάθος, όπου πιθανό να καταλήγαμε αν είχαμε επιλέξει κάποια φάση ανάγνωσης από τη μνήμη), και τέλος επειδή η συγκεκριμένη μέθοδος διευθυνσιοδότησης χρησιμοποιείται συχνά.
- Η τελική διεύθυνση προσπέλασης μνήμης τοποθετείται στον καταχωρητή AR σε κάθε περίπτωση. Έτσι, επιτυγχάνουμε ομοιομορφία στις φάσεις επίλυσης έμμεσης διευθυνσιοδότησης και κατ' ευθείαν προσπέλασης της μνήμης για ανάγνωση σε όλες τις σχετικές εντολές.
- Ο καταχωρητής LR τροφοδοτείται από τον AR στη φάση εκτέλεσης της εντολής LOAD, όταν αυτή προσπελαίνει τη μνήμη, και όχι σε περίπτωση άμεσης διευθυνσιοδότησης.
- Η ΜΔΜ, ο PC και η ΑΔΜ ελέγχονται με τον τρόπο που είδαμε στην προηγούμενη άσκηση.
- Για απλοποίηση, δεν αναγράφουμε σήματα επίτρεψης εγγραφής στους καταχωρητές της ΜΕΔ, όταν η εγγραφή μπορεί να υπονοείται με αναγραφή της επιλογής στην είσοδό τους.
- Τέλος, όπως και στην προηγούμενη άσκηση, κάθε σήμα επιλογής εισόδου κάποιας υπομονάδας ή καταχωρητή που δε χρησιμοποιείται σε κάποιον κύκλο μηχανής γίνεται *αδιάφορη συνθήκη* σε εκείνον τον κύκλο. Κάθε άλλο σήμα που δε χρησιμοποιείται σε κάποιον κύκλο μηχανής πρέπει να είναι *απενεργοποιημένο* σε εκείνον τον κύκλο.

Θεωρώντας και πάλι ως αρχική κατάσταση της ΜΕ την κατάσταση 0, η οποία ενεργοποιεί τις μικρολειτουργίες της φάσης ανάκλησης, η επόμενη κατάσταση 1 ενεργοποιεί τις μικρολειτουργίες

Εντολή	Διευθυνσιοδότηση	Φάση	Μικρολειτουργίες
Φάση Ανάκλησης		0	$\Delta/E = 0$, ανάγνωση, εγγραφή IR, $\text{ΑΛΜ1} = \text{PC}$, $\text{ΑΛΜ2} = 1$, πρόσθεση, $\text{PC} = \text{ΑΛΜ}$
Φάση Αποκωδικοποίησης		1	$\text{ΑΛΜ1} = \Sigma$, $\text{ΑΛΜ2} = \text{PC} \parallel \text{IR}$, πρόσθεση, εγγραφή AR
LOAD	κατ' ευθείαν	2	$\Delta/E = 1$, ανάγνωση, διευθ = $\text{PC} \parallel \text{IR}$, εγγραφή DR, $\text{AR} = \text{PC} \parallel \text{IR}$
		3	$\Sigma = \text{DR}$, $\text{LR} = \text{AR}$
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = $\text{PC} \parallel \text{IR}$, εγγραφή DR, $\text{AR} = \text{PC} \parallel \text{IR}$
		3	$\Delta/E = 1$, ανάγνωση, διευθ = DR, εγγραφή DR, $\text{AR} = \text{DR}$
		4	$\Sigma = \text{DR}$, $\text{LR} = \text{AR}$
	δεικτοδοτούμενη	2	$\Delta/E = 1$, ανάγνωση, διευθ = AR, εγγραφή DR
		3	$\Sigma = \text{DR}$, $\text{LR} = \text{AR}$
	έμμεση δεικτοδοτούμενη	2	$\Delta/E = 1$, ανάγνωση, διευθ = AR, εγγραφή DR
		3	$\Delta/E = 1$, ανάγνωση, διευθ = DR, εγγραφή DR, $\text{AR} = \text{DR}$
		4	$\Sigma = \text{DR}$, $\text{LR} = \text{AR}$
άμεση	2	$\Sigma = \text{προεκτ}(\text{IR})$	
STORE	κατ' ευθείαν	2	$\Delta/E = 1$, εγγραφή, διευθ = $\text{PC} \parallel \text{IR}$
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = $\text{PC} \parallel \text{IR}$, εγγραφή DR, $\text{AR} = \text{PC} \parallel \text{IR}$
		3	$\Delta/E = 1$, εγγραφή, διευθ = DR
	υπονοούμενη	2	$\Delta/E = 1$, εγγραφή, διευθ = LR
ADD	κατ' ευθείαν	2	$\Delta/E = 1$, ανάγνωση, διευθ = $\text{PC} \parallel \text{IR}$, εγγραφή DR, $\text{AR} = \text{PC} \parallel \text{IR}$
		3	$\text{ΑΛΜ1} = \Sigma$, $\text{ΑΛΜ2} = \text{DR}$, πρόσθεση, $\Sigma = \text{ΑΛΜ}$
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = $\text{PC} \parallel \text{IR}$, εγγραφή DR, $\text{AR} = \text{PC} \parallel \text{IR}$
		3	$\Delta/E = 1$, ανάγνωση, διευθ = DR, εγγραφή DR, $\text{AR} = \text{DR}$
		4	$\text{ΑΛΜ1} = \Sigma$, $\text{ΑΛΜ2} = \text{DR}$, πρόσθεση, $\Sigma = \text{ΑΛΜ}$
	άμεση	2	$\text{ΑΛΜ1} = \Sigma$, $\text{ΑΛΜ2} = \text{προεκτ}(\text{IR})$, πρόσθεση, $\Sigma = \text{ΑΛΜ}$
SUB	κατ' ευθείαν	2	$\Delta/E = 1$, ανάγνωση, διευθ = $\text{PC} \parallel \text{IR}$, εγγραφή DR, $\text{AR} = \text{PC} \parallel \text{IR}$
		3	$\text{ΑΛΜ1} = \Sigma$, $\text{ΑΛΜ2} = \text{DR}$, αφαίρεση, $\Sigma = \text{ΑΛΜ}$
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = $\text{PC} \parallel \text{IR}$, εγγραφή DR, $\text{AR} = \text{PC} \parallel \text{IR}$
		3	$\Delta/E = 1$, ανάγνωση, διευθ = DR, εγγραφή DR, $\text{AR} = \text{DR}$
		4	$\text{ΑΛΜ1} = \Sigma$, $\text{ΑΛΜ2} = \text{DR}$, αφαίρεση, $\Sigma = \text{ΑΛΜ}$
JMP	κατ' ευθείαν	2	$\text{PC} = \text{PC} \parallel \text{IR}$
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = $\text{PC} \parallel \text{IR}$, εγγραφή DR, $\text{AR} = \text{PC} \parallel \text{IR}$
		3	$\text{PC} = \text{DR}$
	δεικτοδοτούμενη	2	$\text{PC} = \text{AR}$
	έμμεση δεικτοδοτούμενη	2	$\Delta/E = 1$, ανάγνωση, διευθ = AR, εγγραφή DR
	3	$\text{PC} = \text{DR}$	
JAL	κατ' ευθείαν	2	$\text{PC} = \text{PC} \parallel \text{IR}$, $\Sigma = \text{PC}$
	έμμεση	2	$\Delta/E = 1$, ανάγνωση, διευθ = $\text{PC} \parallel \text{IR}$, εγγραφή DR, $\text{AR} = \text{PC} \parallel \text{IR}$
		3	$\text{PC} = \text{DR}$, $\Sigma = \text{PC}$
	δεικτοδοτούμενη	2	$\text{PC} = \text{AR}$, $\Sigma = \text{PC}$
	έμμεση δεικτοδοτούμενη	2	$\Delta/E = 1$, ανάγνωση, διευθ = AR, εγγραφή DR
3	$\text{PC} = \text{DR}$, $\Sigma = \text{PC}$		
BNZ	σχετική	2	$\text{ΑΛΜ1} = \text{PC}$, $\text{ΑΛΜ2} = \text{προεκτ}(\text{IR})$, πρόσθεση, if ($\Sigma \neq 0$) $\text{PC} = \text{ΑΛΜ}$
BLTZ	σχετική	2	$\text{ΑΛΜ1} = \text{PC}$, $\text{ΑΛΜ2} = \text{προεκτ}(\text{IR})$, πρόσθεση, if ($\Sigma < 0$) $\text{PC} = \text{ΑΛΜ}$

τουργίες της φάσης αποκωδικοποίησης. Από αυτήν, λαμβάνοντας υπ' όψη τις κοινές φάσεις, οδηγούμαστε σε μία από 11 καταστάσεις. Χρησιμοποιώντας τον AR με τον τρόπο που αναφέραμε, η φάση επίλυσης έμμεσης και έμμεσης δεικτοδοτούμενης διευθυνσιοδότησης αποτελούν κοινές φάσεις για όλες τις σχετικές εντολές. Η τελική προσπέλαση ανάγνωσης από τη μνήμη είναι κοινή φάση, ανάλογα με την πηγή της εισόδου διευθύνσεων της ΜΑΜ. Ακόμα, τόσο η κατ' ευθείαν, όσο και η έμμεση διευθυνσιοδότηση στις εντολές ADD και SUB οδηγούν σε κοινή φάση εκτέλεσης. Τέλος, οι δύο εντολές άλματος με συνθήκη έχουν επίσης κοινή φάση εκτέλεσης. Συνεχίζοντας την ανάλυση καταστάσεων για κάθε εντολή και μέθοδο



διευθυνσιοδότησης που αυτή υποστηρίζει, κατασκευάζουμε τη μηχανή καταστάσεων που περιγράφεται από το παραπάνω διάγραμμα.

Θα δώσουμε στη συνέχεια τον αντίστοιχο πίνακα μετάβασης καταστάσεων. Θεωρούμε ότι οι κωδικοί λειτουργίας Op των εντολών που υποστηρίζονται παίρνουν τις τιμές που είχαμε στην προηγούμενη άσκηση. Ακόμα, θεωρούμε ότι τα ψηφία διευθυνσιοδότησης $\Delta \equiv IR^{28-23}$ αντιστοιχούν στις μεθόδους διευθυνσιοδότησης με τη σειρά: κατ' ευθείαν, άμεση, έμμεση, δεικτοδοτούμενη, υπονοούμενη και σχετική, με το ψηφίο Δ_0 να υποδηλώνει κατ' ευθείαν διευθυνσιοδότηση, το Δ_1 να υποδηλώνει άμεση διευθυνσιοδότηση, κοκ. Η μόνη περίπτωση που πολλαπλά ψηφία διευθυνσιοδότησης έχουν τιμή 1 είναι η έμμεση δεικτοδοτημένη διευθυνσι-

οδότηση, για την οποία θα έχουμε $\Delta=001100$. Σε όλες τις άλλες περιπτώσεις μόνο ένα ψηφίο διευθυνσιοδότησης επιτρέπεται να έχει τιμή 1.

Αναλύοντας τις μεταβάσεις καταστάσεων για κάθε εντολή, βρίσκουμε τις συνθήκες μετάβασης $C(i,j)$ από μια κατάσταση i σε μια άλλη j , όπως δίνονται στον πίνακα που ακολουθεί:

Συνθήκη Μετάβασης	Τιμή
C(1,2)	$\Delta_3' * \Delta_2 + \Delta_0 * ((Op==LOAD) + (Op==ADD) + (Op==SUB))$
C(1,5)	$\Delta_3 * ((Op==LOAD) + \Delta_2 * ((Op==JMP) + (Op==JAL)))$
C(1,6)	$\Delta_1 * (Op==LOAD)$
C(1,7)	$\Delta_0 * (Op==STORE)$
C(1,9)	$\Delta_4 * (Op==STORE)$
C(1,11)	$\Delta_1 * (Op==ADD)$
C(1,12)	$\Delta_0 * (Op==JMP)$
C(1,14)	$\Delta_3 * \Delta_2' * (Op==JMP)$
C(1,15)	$\Delta_0 * (Op==JAL)$
C(1,17)	$\Delta_3 * \Delta_2' * (Op==JAL)$
C(1,18)	$\Delta_5 * ((Op==BNZ) + (Op==BLTZ))$
C(2,3)	$\Delta_0 * (Op==LOAD)$
C(2,4)	$\Delta_2 * ((Op==LOAD) + (Op==ADD) + (Op==SUB))$
C(2,8)	$(Op==STORE)$
C(2,10)	$\Delta_0 * ((Op==ADD) + (Op==SUB))$
C(2,13)	$(Op==JMP)$
C(2,16)	$(Op==JAL)$
C(4,3)	$(Op==LOAD)$
C(4,10)	$((Op==ADD) + (Op==SUB))$
C(5,3)	$\Delta_2' * (Op==LOAD)$
C(5,4)	$\Delta_2 * (Op==LOAD)$
C(5,13)	$(Op==JMP)$
C(5,16)	$(Op==JAL)$

Για παράδειγμα, η μετάβαση από την κατάσταση 1 στην κατάσταση 2 γίνεται όταν έχουμε έμμεση (και όχι δεικτοδοτούμενη) διευθυνσιοδότηση, ή όταν έχουμε κατ' ευθείαν διευθυνσιοδότηση σε εντολή που έχει τελική προσπέλαση ανάγνωσης στη μνήμη, δηλαδή μια από τις LOAD, ADD και SUB.

Όταν έχουμε δεικτοδοτούμενη διευθυνσιοδότηση, μεταβαίνουμε από την κατάσταση 1 στην κατάσταση 5, εκτός εάν έχουμε εντολή άλματος JMP ή JAL χωρίς έμμεση διευθυνσιοδότηση. Σε αυτές τις δύο περιπτώσεις δε χρειαζόμαστε προσπέλαση στη μνήμη, οπότε μεταβαίνουμε αντίστοιχα στις καταστάσεις 14 ή 17.

Από την κατάσταση 2 μεταβαίνουμε στην κατάσταση 4 μόνο όταν έχουμε έμμεση διευθυνσιοδότηση, και εφ' όσον δεν έχουμε εντολή STORE ούτε μια από τις εντολές άλματος JMP ή JAL. Όταν είμαστε στην κατάσταση 2 με εντολή STORE, έχουμε ήδη εξασφαλίσει τη συνθήκη έμμεσης διευθυνσιοδότησης, κι επομένως μεταβαίνουμε απ' ευθείας στην κατάσταση 8. Το ίδιο συμβαίνει για τις εντολές JMP και JAL, οι οποίες οδηγούν αντίστοιχα στις καταστάσεις 13 και 16.

Η κατάσταση 4 αντιστοιχεί στην περίπτωση που έχουμε έμμεση (ή έμμεση δεικτοδοτούμενη) διευθυνσιοδότηση με τελική προσπέλαση ανάγνωσης στη μνήμη, έχοντας ήδη υπολογίσει τη διεύθυνση τελικής προσπέλασης. Αυτό θα συμβαίνει για μια από τις εντολές LOAD, ADD ή SUB. Εκτελούμε την τελική προσπέλαση και στην περίπτωση της LOAD μεταβαίνουμε στην κατάσταση 3, διαφορετικά στην κατάσταση 10.

Στην κατάσταση 5 τέλος θα βρεθούμε όταν έχουμε δεικτοδοτούμενη διευθυνσιοδότηση στην οποία χρειαζόμαστε προσπέλαση στη μνήμη. Η προσπέλαση αυτή είναι είτε τελική προσπέλαση ανάγνωσης είτε προσπέλαση για επίλυση έμμεσης διευθυνσιοδότησης. Προσπελάουμε τη μνήμη στη διεύθυνση που έχουμε αποθηκεύσει στον AR, και εάν έχουμε μία από τις εντολές άλματος JMP και JAL (που σημαίνει ότι έχουμε έμμεση δεικτοδοτούμενη διευθυνσιοδότηση χωρίς τελική προσπέλαση στη μνήμη), μεταβαίνουμε απ' ευθείας στην αντίστοιχη κατάσταση 13 ή 16. Για την περίπτωση εντολής LOAD, μπορεί να έχουμε είτε έμμεση δει-

κτοδοτούμενη διευθυνσιοδότηση, οπότε η προσπέλαση που κάναμε αποτελούσε επίλυση έμμεσης αναφοράς και συνεπώς χρειαζόμαστε την τελική προσπέλαση στη μνήμη, είτε απλή δεικτοδοτούμενη διευθυνσιοδότηση, οπότε η προσπέλαση που κάναμε αποτελούσε την τελική προσπέλαση που μας επέστρεψε το ζητούμενο δεδομένο. Στην πρώτη από τις δύο αυτές περιπτώσεις μεταβαίνουμε στην κατάσταση 4, ενώ στη δεύτερη μεταβαίνουμε στην κατάσταση 3.

Οι μεταβάσεις που δεν υπάρχουν στον πίνακα γίνονται με συνθήκη 1.

Στη συνέχεια θα βρούμε τις λογικές συναρτήσεις που εκφράζουν τις επόμενες καταστάσεις και τα σήματα ελέγχου της αρχιτεκτονικής μας, σε συνδυασμό των τιμών των εισόδων και των παρουσών καταστάσεων. Οι εισοδοί είναι ο κωδικός λειτουργίας και τα ψηφία διευθυνσιοδότησης της λέξης εντολής.

Αν χρησιμοποιούμε την τεχνική ενός ενεργού σήματος για την κωδικοποίηση της λέξης κατάστασης, από τον παραπάνω πίνακα θα πάρουμε:

$$\begin{aligned}
 S_2 &= S_1 (\Delta_3 \Delta_2 + \Delta_0 (Op_2' Op_1' Op_0' + Op_2' Op_1 Op_0' + Op_2' Op_1 Op_0)) = \\
 &= S_1 (\Delta_3 \Delta_2 + \Delta_0 Op_2' (Op_0' + Op_1)) \\
 S_3 &= S_2 (\Delta_0 Op_2' Op_1' Op_0') + S_4 (Op_2' Op_1' Op_0') + S_5 (\Delta_2' Op_2' Op_1' Op_0') = \\
 &= (S_2 \Delta_0 + S_4 + S_5 \Delta_2') Op_2' Op_1' Op_0' \\
 S_4 &= S_2 (\Delta_2 (Op_2' Op_1' Op_0' + Op_2' Op_1 Op_0' + Op_2' Op_1 Op_0)) + S_5 (\Delta_2 Op_2' Op_1' Op_0') = \\
 &= \Delta_2 Op_2' (S_2 (Op_0' + Op_1) + S_5 Op_1' Op_0') \\
 S_5 &= S_1 (\Delta_3 (Op_2' Op_1' Op_0' + \Delta_2 (Op_2 Op_1' Op_0' + Op_2 Op_1' Op_0))) = \\
 &= S_1 \Delta_3 Op_1' (Op_2' Op_0' + \Delta_2 Op_2) \\
 S_6 &= S_1 \Delta_1 Op_2' Op_1' Op_0' \\
 S_7 &= S_1 \Delta_0 Op_2' Op_1' Op_0 \\
 S_8 &= S_2 Op_2' Op_1' Op_0 \\
 S_9 &= S_1 \Delta_4 Op_2' Op_1' Op_0 \\
 S_{10} &= S_2 (\Delta_0 (Op_2' Op_1 Op_0' + Op_2' Op_1 Op_0)) + S_4 (Op_2' Op_1 Op_0' + Op_2' Op_1 Op_0) = \\
 &= (S_2 \Delta_0 + S_4) Op_2' Op_1 \\
 S_{11} &= \Delta_1 Op_2' Op_1 Op_0' \\
 S_{12} &= \Delta_0 Op_2 Op_1' Op_0' \\
 S_{13} &= (S_2 + S_5) Op_2 Op_1' Op_0' \\
 S_{14} &= S_1 \Delta_3 \Delta_2' Op_2 Op_1' Op_0' \\
 S_{15} &= \Delta_0 Op_2 Op_1' Op_0 \\
 S_{16} &= (S_2 + S_5) Op_2 Op_1' Op_0 \\
 S_{17} &= S_1 \Delta_3 \Delta_2' Op_2 Op_1' Op_0 \\
 S_{18} &= S_1 \Delta_5 (Op_2 Op_1 Op_0' + Op_2 Op_1 Op_0) = S_1 \Delta_5 Op_2 Op_1
 \end{aligned}$$

και για τις υπόλοιπες μεταβάσεις:

$$\begin{aligned}
 S_1 &= S_0 \\
 S_0 &= S_3 + S_6 + S_7 + S_8 + S_9 + S_{10} + S_{11} + S_{12} + S_{13} + S_{14} + S_{15} + S_{16} + S_{17} + S_{18}
 \end{aligned}$$

Όσο αφορά τα σήματα ελέγχου της αρχιτεκτονικής, προχωράμε όπως ακριβώς κάναμε στην προηγούμενη άσκηση, οπότε βρίσκουμε:

$$\begin{aligned}
 \text{“ανάγνωση”} &= S_0 + S_2 + S_4 + S_5 \\
 \text{“εγγραφή”} &= S_7 + S_8 + S_9 \\
 \text{“Δ/Ε”} &= S_2 + S_4 + S_5 + S_7 + S_8 + S_9 \\
 \text{“εγγραφή } \Sigma \text{”} &= S_3 + S_6 + S_{10} + S_{11} + S_{15} + S_{16} + S_{17} \\
 \text{“εγγραφή AR”} &= S_1 + S_2 + S_4 \\
 \text{“εγγραφή DR”} &= S_2 + S_4 + S_5 \\
 \text{“εγγραφή IR”} &= S_0 \\
 \text{“εγγραφή LR”} &= S_3 \\
 \text{“εγγραφή PC”} &= S_0 + S_{12} + S_{13} + S_{14} + S_{15} + S_{16} + S_{17} + S_{18} \text{ “συνθ}(\Sigma)\text{”} \\
 \text{“πρόσθεση”} &= S_0 + S_1 + S_{10} \text{ “πράξη(ME)”} + S_{11} + S_{18}
 \end{aligned}$$

Στην παρούσα άσκηση, τα σήματα επιλογής εισόδων των υπομονάδων της ΜΕΔ διαμορφώνονται ανάλογα με τις τροποποιήσεις που έχουμε κάνει.

Τα ψηφία επιλογής του πολυπλέκτη ΜΣ για την είσοδο στον Σ, με σειρά επιλογής από την ΑΛΜ, τον ΡC, τον ΙR (με προέκταση προσήμου) ή τον DR, θα είναι:

$$“ΜΣ_1” = S_6 + S_3$$

$$“ΜΣ_0” = S_{15} + S_{16} + S_{17} + S_3$$

Η είσοδος διευθύνσεων της ΜΔΜ μπορεί τώρα να γίνει από τον ΡC, τον ΙR (με παράθεση ψηφίων του ΡC), τον AR, τον DR ή τον LR. Προκειμένου να αποφύγουμε τη χρήση ενός πολυπλέκτη 8×1, χρησιμοποιούμε δύο μικρότερους, έναν πολυπλέκτη 2×1 που επιλέγει μεταξύ του ΡC και της εξόδου ενός δεύτερου πολυπλέκτη 4×1, ο οποίος επιλέγει μεταξύ των υπόλοιπων καταχωρητών. Αν ο πρώτος πολυπλέκτης επιλέγει τον ΡC με τιμή επιλογής 0, το σήμα επιλογής του θα είναι απλά το “Δ/Ε”. Όσο αφορά το δεύτερο πολυπλέκτη, αν οι επιλογές των εισόδων του κωδικοποιούνται με τη σειρά που αναφέραμε, το σήμα επιλογής του “διευθ” θα είναι:

$$“διευθ_1” = S_4 + S_8 + S_9$$

$$“διευθ_0” = S_5 + S_9$$

Στην ΑΛΜ, η είσοδος ΑΛΜ1 επιλέγει μεταξύ του Σ και του ΡC, οπότε γι’ αυτή τη σειρά:

$$“ΑΛΜ1” = S_0 + S_{18}$$

Η είσοδος ΑΛΜ2 επιλέγει μεταξύ του DR, του ΙR – είτε με προέκταση προσήμου των ψηφίων ΙR²²⁻⁰ είτε με παράθεση αυτών με τα 9 πιο σημαντικά ψηφία του ΡC – και της σταθεράς 1. Μπορούμε να χρησιμοποιήσουμε έναν πολυπλέκτη 4×1 για την επιλογή αυτή, οπότε αν οι επιλογές των εισόδων του κωδικοποιούνται με τη σειρά που αναφέρθηκαν, τότε:

$$“ΑΛΜ2_1” = S_0 + S_1$$

$$“ΑΛΜ2_0” = S_0 + S_{11} + S_{18}$$

Ο ΡC δέχεται είσοδο από τον ΙR (με παράθεση ψηφίων του ΡC), από τον DR, από τον AR και από την ΑΛΜ. Έτσι, ο έλεγχος του αντίστοιχου πολυπλέκτη 4×1 MPC, με κωδικοποίηση εισόδων σύμφωνα με τη σειρά που αυτές αναφέρθηκαν, θα γίνεται ως εξής:

$$“MPC_1” = S_0 + S_{14} + S_{17} + S_{18}$$

$$“MPC_0” = S_0 + S_{13} + S_{16} + S_{18}$$

Τέλος, ο AR δέχεται είσοδο από τον ΙR (με παράθεση ψηφίων του ΡC), από τον DR, και από την ΑΛΜ. Ο έλεγχος του αντίστοιχου πολυπλέκτη 4×1 MAR, με κωδικοποίηση εισόδων με τη σειρά που τις αναφέραμε, θα γίνεται:

$$“MAR_1” = S_1$$

$$“MAR_0” = S_4$$

Οι λογικές εκφράσεις των σημάτων “πράξη(ME)” και “συνθ(Σ)” παραμένουν οι ίδιες με τις εκφράσεις της προηγούμενης άσκησης:

$$“πράξη(ME)” = Op_2’Op_1Op_0’$$

με την υπόθεση ότι η πρόσθεση κωδικοποιείται στον αριθμό 1, και:

$$“συνθ(Σ)” = Z’Op_2Op_1Op_0’ + NOp_2Op_1Op_0$$