

Πανεπιστήμιο Θεσσαλίας
Τμήμα Πληροφορικής

Οργάνωση Η/Υ

Ενότητα 2η: Μονάδα Ελέγχου Απλού Κύκλου Μηχανής

Άσκηση 1:

Θεωρήστε τη μονάδα επεξεργασίας δεδομένων της απλοποιημένης αρχιτεκτονικής MIPS, στην οποία κάθε εντολή ολοκληρώνεται σε έναν κύκλο μηχανής. Σχεδιάστε την υπομονάδα ελέγχου της ΑΛΜ, όταν οι δυνατές εντολές της αρχιτεκτονικής, οι αντίστοιχοι κωδικοί λειτουργίας που προωθούνται στην ΑΛΜ από την κεντρική ΜΕ, οι αντίστοιχοι κωδικοί τελεστή και οι αντίστοιχες μικρολειτουργίες που πρέπει να εκτελεί η ΑΛΜ δίνονται στον πιο κάτω πίνακα:

Εντολή	Κωδικός Λειτουργίας ΑΛΜ	Κωδικός Τελεστή	Μικρολειτουργία ΑΛΜ
<i>lw</i>	00	xxxxxx	<i>add</i>
<i>sw</i>	00	xxxxxx	<i>add</i>
<i>beq</i>	01	xxxxxx	<i>sub</i>
<i>add</i>	10	100000	<i>add</i>
<i>sub</i>	10	100010	<i>sub</i>
<i>and</i>	10	100100	<i>and</i>
<i>or</i>	10	100101	<i>or</i>
<i>slt</i>	10	101010	<i>slt</i>

Απάντηση:

Η ΑΛΜ που μελετάμε υποστηρίζει 5 διαφορετικές μικρολειτουργίες. Η υπομονάδα ελέγχου της ΑΛΜ δέχεται από την κεντρική μονάδα ελέγχου της ΚΜΕ μετά την αποκωδικοποίηση της εντολής δύο ψηφία που αντιστοιχούν στον κωδικό λειτουργίας ΑΛΜ και τα 6 ψηφία του κωδικού τελεστή. Μπορούμε να επιλέξουμε μία από τις εξής δύο μορφές εξόδου της ζητούμενης μονάδας ελέγχου:

(α) Η μονάδα ελέγχου της ΑΛΜ παράγει 5 ανεξάρτητες γραμμές ελέγχου, που καθεμιά ενεργοποιεί μία μικρολειτουργία. Στην περίπτωση αυτή, σε κάθε κύκλο μηχανής μόνο μία από τις 5 γραμμές μπορεί να έχει τη λογική τιμή 1.

(β) Η μονάδα ελέγχου παράγει έναν κωδικοποιημένο αριθμό μικρολειτουργίας. Αυτός τροφοδοτεί έναν πολυπλέκτη της ΑΛΜ που επιλέγει την τελική πράξη που εκτελείται.

Στις περισσότερες περιπτώσεις που έχουμε να επιλέξουμε μεταξύ μικρολειτουργιών της ίδιας υπομονάδας της ΚΜΕ, είναι προτιμότερο να στέλνουμε σε αυτήν έναν κωδικοποιημένο αριθμό μικρολειτουργίας σύμφωνα με την (β) πιο πάνω τεχνική, παρά να στέλνουμε ανεξάρτητες γραμμές ελέγχου σύμφωνα με την (α) τεχνική. Όταν μια υπομονάδα όπως η ΑΛΜ εκτελεί μερικές δεκάδες μικρολειτουργίες – κι όχι τις 5 που έχουμε στο απλουστευμένο παράδειγμά μας, είναι πολύ πιο εύκολο να διαδώσουμε πάνω στην επιφάνεια του κυκλώματος που υλοποιεί την ΚΜΕ έναν κώδικα μικρολειτουργίας μέσω ενός μικρού αριθμού γραμμών παρά μερικές δεκάδες σήματα ελέγχου μέσω ισάριθμων ανεξάρτητων γραμμών.

Επιλέγοντας λοιπόν τη (β) από τις πιο πάνω τεχνικές, μπορούμε να κωδικοποιήσουμε τις μικρολειτουργίες ως εξής:

Μικρολειτουργία	Κωδικός
add	010
sub	110
and	000
or	001
slt	111

Ο υπολογισμός της τιμής κάθε ψηφίου του κωδικού μικρολειτουργίας γίνεται μέσω κάποιας λογικής συνάρτησης. Προκειμένου να βρούμε τις τρεις λογικές συναρτήσεις που θέλουμε για τα τρία ψηφία, φτιάχνουμε τους αντίστοιχους πίνακες αλήθειας. Ξεκινάμε με έναν συνολικό πίνακα αλήθειας που προκύπτει άμεσα από τον πίνακα που μας δίνεται, και στη συνέχεια απομονώνουμε τις περιπτώσεις που δίνουν τη λογική τιμή 1 ξεχωριστά για κάθε ψηφίο εξόδου, έτσι ώστε να πάρουμε για καθένα μια όσο πιο απλοποιημένη λογική παράσταση γίνεται, λαμβάνοντας υπ' όψη τις αδιάφορες συνθήκες (ή μη χρησιμοποιούμενες τιμές) των σημάτων εισόδου. Ο λόγος που παίρνουμε κάθε ψηφίο ξεχωριστά είναι η διαφορετική συμπεριφορά των αδιάφορων συνθηκών σε κάθε μία περίπτωση, όπως θα δούμε παρακάτω.

Ονομάζοντας τα ψηφία εισόδου ALUOp1, ALUOp0, F5, F4, F3, F2, F1 και F0 και τα ψηφία εξόδου Oper2, Oper1 και Oper0, ο συνολικός πίνακας αλήθειας είναι:

ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	Oper2	Oper1	Oper0
0	0	x	x	x	x	x	x	0	1	0
0	1	x	x	x	x	x	x	1	1	0
1	0	x	x	0	0	0	0	0	1	0
1	0	x	x	0	0	1	0	1	1	0
1	0	x	x	0	1	0	0	0	0	0
1	0	x	x	0	1	0	1	0	0	1
1	0	x	x	1	0	1	0	1	1	1

όπου έχουμε θέσει τα δύο πρώτα ψηφία από τον κωδικό τελεστή αδιάφορες συνθήκες, εφ' όσον έχουν για όλες τις περιπτώσεις είτε τις ίδιες είτε αδιάφορες τιμές. Τα ψηφία αυτά θα μπορούσαν και να παραλειφθούν.

Από τον παραπάνω πίνακα κατασκευάζουμε τρεις νέους πίνακες, παίρνοντας για κάθε ψηφίο εξόδου τις περιπτώσεις που δίνουν λογική τιμή 1. Ο κάθε πίνακας μετατρέπει σε αδιάφορες συνθήκες τιμές που δεν παίζουν ρόλο στην τιμή της εξόδου. Αυτό επιτυγχάνεται, αν υποθέσουμε ότι συνδυασμοί ψηφίων εισόδου που δεν υπάρχουν στον αρχικό πίνακα αλήθειας δεν πρόκειται να εμφανιστούν. Θεωρώντας αυτούς τους συνδυασμούς αδιάφορες συνθήκες για την έξοδο, διατηρούμε μόνο τις τιμές που είναι καθοριστικές για τη λογική τιμή 1 της κάθε εξόδου.

Έτσι, για το ψηφίο Oper2 λαμβάνουμε τον πίνακα:

ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	Έξοδος
x	1	x	x	x	x	x	x	Oper2 = 1
1	x	x	x	x	x	1	x	

Η πρώτη γραμμή προκύπτει παρατηρώντας ότι όταν το ψηφίο ALUOp0 έχει τιμή 1, το ψηφίο Oper2 θα λάβει τιμή 1, ανεξάρτητα από τις τιμές των υπόλοιπων ψηφίων εισόδου. Παρόμοια, η δεύτερη τιμή προκύπτει παρατηρώντας ότι όταν τα ψηφία ALUOp1 και F1 έχουν τιμή 1, το ψηφίο Oper2 θα λάβει πάλι τιμή 1, ανεξάρτητα από τις υπόλοιπες τιμές. Εδώ δεν αρκεί το ψηφίο F1, παρ' όλο που θα μπορούσε κανείς να ξεγελαστεί από τις τιμές του ψηφίου αυτού – τιμές 1 όταν το Oper2 γίνεται 1, επειδή τιμή 1 για το ψηφίο αυτό μπορεί να εμφανιστεί και στην πρώτη γραμμή του αρχικού πίνακα αλήθειας, όπου είναι αδιάφορη συνθήκη, και όπου όμως το Oper2 δε λαμβάνει τιμή 1!

Για το ψηφίο Oper1 ο αντίστοιχος πίνακας θα είναι:

ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	Έξοδος
0	x	x	x	x	x	x	x	Oper1 = 1
1	x	x	x	x	0	x	x	

Εδώ η πρώτη γραμμή προκύπτει επειδή όταν το ψηφίο ALUOp1 έχει τιμή 0, η έξοδος Oper1 θα έχει πάντα τιμή 1. Η δεύτερη γραμμή προκύπτει επειδή όταν το ψηφίο ALUOp1 έχει τιμή 1 και το ψηφίο F2 έχει τιμή 0, η έξοδος Oper1 έχει τιμή 1, ανεξάρτητα από τις υπόλοιπες τιμές εισόδου. Μάλιστα, λόγω της πρώτης γραμμής, μπορούμε να μετατρέψουμε σε αδιάφορη συνθήκη και την είσοδο ALUOp1 της δεύτερης γραμμής, κάτι που θα προκύψει και πιο κάτω. Τέλος, για το ψηφίο Oper0 λαμβάνουμε τον πίνακα:

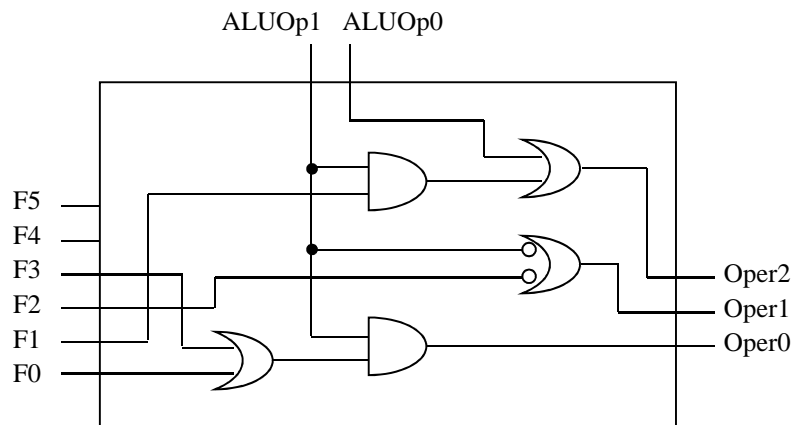
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	Έξοδος
1	x	x	x	x	x	x	1	Oper0 = 1
1	x	x	x	1	x	x	x	

Στην περίπτωση αυτή η είσοδος ALUOp0 γίνεται πάλι αδιάφορη συνθήκη όταν το ψηφίο ALUOp1 έχει τιμή 1. Η πρώτη γραμμή προκύπτει, επειδή η έξοδος Oper0 λαμβάνει τιμή 1, όταν το ψηφίο F0 έχει τιμή 1, κάνοντας αδιάφορες συνθήκες όλες τις άλλες εισόδους του κωδικού τελεστή. Παρόμοια προκύπτει η δεύτερη γραμμή για την τιμή F3 = 1.

Σύμφωνα με τα παραπάνω, οι λογικές συναρτήσεις που μας δίνουν τα ψηφία εξόδου της ζητούμενης μονάδας ελέγχου θα είναι:

$$\begin{aligned} \text{Oper2} &= \text{ALUOp0} + \text{ALUOp1} * \text{F1} \\ \text{Oper1} &= \text{ALUOp1}' + \text{ALUOp1} * \text{F2}' = \text{ALUOp1}' + \text{F2}' \\ \text{Oper0} &= \text{ALUOp1} * (\text{F0} + \text{F3}) \end{aligned}$$

Η ζητούμενη μονάδα ελέγχου θα είναι επομένως:



Η παραπάνω μορφή των λογικών συναρτήσεων δεν είναι και η μοναδική που υλοποιεί τον πίνακα αλήθειας των σημάτων ελέγχου. Πρέπει να τονίσουμε ότι η μορφή αυτή προέκυψε, επειδή θεωρήσαμε ότι διαφορετικοί συνδυασμοί τιμών εισόδου δεν πρόκειται να εμφανιστούν. Αυτό είναι σωστό, μόνο αν υποθέσουμε είτε ότι δε μας ενδιαφέρει η ανίχνευση μη αποδεκτών εντολών, είτε ότι η κεντρική μονάδα ελέγχου της ΚΜΕ ανιχνεύει πλήρως τις μη αποδεκτές εντολές. Έτσι, μπορούμε να κάνουμε τις πιο πάνω απλοποιήσεις χωρίς να κινδυνεύουμε να παράγουμε λανθασμένη μονάδα ελέγχου ΑΛΜ. Διαφορετικά, θα πρέπει να επιτρέψουμε την εμφάνιση και διαφορετικών τιμών εισόδου, οπότε αυτές δε μετατρέπονται σε αδιάφορες συνθήκες, με πιθανό αποτέλεσμα την ενεργοποίηση ειδικής περίπτωσης στην ΚΜΕ για αυτές τις τιμές. Ένα μέσο σενάριο είναι να θεωρήσουμε ότι η κεντρική μονάδα ε-

λέγχου δεν παράγει ποτέ μη αποδεκτές τιμές στα ψηφία κωδικού λειτουργίας ΑΛΜ, αλλά ότι προωθεί τα ψηφία του κωδικού τελεστή από τη λέξη εντολής στη μονάδα ελέγχου ΑΛΜ χωρίς να τα ελέγχει. Σε τέτοια περίπτωση, η σχεδιάσή μας θα πρέπει να επιτρέπει απλοποιήσεις με βάση τα πρώτα, αλλά πιθανά όχι με βάση τα δεύτερα ψηφία.

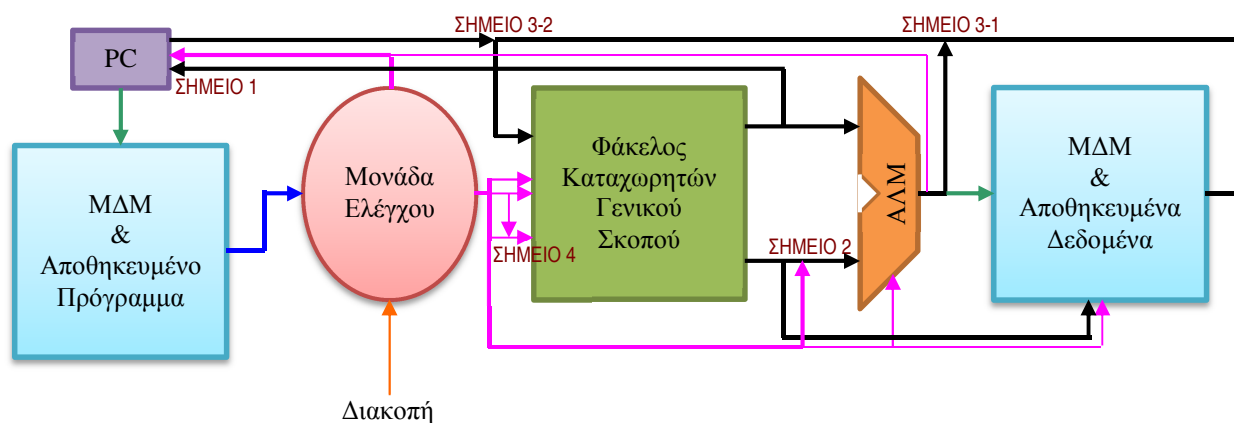
Γενικά στα σύγχρονα συστήματα η τελική μορφή των λογικών συναρτήσεων και των αντίστοιχων κυκλωμάτων ελέγχου, λαμβάνοντας υπόψη και το παραπάνω σχόλιο, προκύπτει αυτόματα μέσα από εργαλεία CAD. Οι όποιοι περιορισμοί παρέχονται με κάποια κατάλληλη κωδικοποίηση, και τα εργαλεία αυτά προσπαθούν να ελαχιστοποιήσουν (α) τον αριθμό των πυλών που απαιτούνται, (β) το χρόνο διάδοσης των σημάτων, αλλά και (γ) την καταναλισκόμενη ισχύ. Σε πραγματικές υλοποιήσεις, όπου οι πίνακες αλήθειας είναι σημαντικά πιο πολύπλοκοι, η βελτιστοποίηση στην σχεδίαση των συναρτήσεων και των κυκλωμάτων δεν είναι απλή διαδικασία, ούτε έχει μονοσήμαντη λύση. Με την εξέλιξη της τεχνολογίας, οι απαιτήσεις αυξάνονται και οι περιορισμοί γίνονται όλο και πιο σύνθετοι. Έτσι, δεν είναι πια εφικτή η σχεδίαση με το χέρι, και η χρήση εργαλείων CAD καθίσταται επιβεβλημένη.

Άσκηση 2:

Θεωρήστε τη μονάδα επεξεργασίας δεδομένων της απλοποιημένης αρχιτεκτονικής MIPS, στην οποία κάθε εντολή ολοκληρώνεται σε έναν κύκλο μηχανής. Μελετήστε τη ροή πληροφορίας στη μονάδα και βρείτε ποιες υπομονάδες δέχονται στην ίδια είσοδο τιμή από περισσότερες της μιας πηγές. Σε κάθε περίπτωση εξηγήστε με λίγα λόγια το μηχανισμό επιλογής, καθώς και με ποια σήματα ελέγχου η μονάδα θα επιλέγει την τιμή από τη σωστή πηγή. Περιγράψτε τα σήματα ελέγχου κι εξηγήστε πώς θα τα παράγετε.

Απάντηση:

Θεωρούμε το διάγραμμα της μονάδας επεξεργασίας δεδομένων (datapath – ΜΕΔ) της συγκεκριμένης απλοποιημένης αρχιτεκτονικής MIPS του ενός κύκλου μηχανής ανά κύκλο εντολής:



Η πληροφορία στη μονάδα αυτή διαδίδεται ως εξής:

(α) Η μονάδα μετρητή προγράμματος (PC) στέλνει στη μονάδα διαχείρισης μνήμης (ΜΑΜ) εντολών όπου βρίσκεται το αποθηκευμένο πρόγραμμα τη διεύθυνση από την οποία η τελευταία θα διαβάσει μια λέξη εντολής. Σε εντολές άλματος με σύνδεση στέλνει τη διεύθυνση επόμενης εντολής στο φάκελο καταχωρητών γενικού σκοπού.

(β) Η μνήμη εντολών εκτελεί την ανάγνωση και στέλνει τη λέξη εντολής στην κεντρική μονάδα ελέγχου.

(γ) Η κεντρική μονάδα ελέγχου αποκωδικοποιεί την εντολή και στέλνει σήματα ελέγχου που πιθανά περιλαμβάνουν ολόκληρες ομάδες bits από την εντολή στις διάφορες υπομονάδες της ΜΕΔ. Μια εξωτερική διακοπή μπορεί να παρέμβει και να αλλοιώσει τη ροή προγράμματος, αναγκάζοντας τη μονάδα ελέγχου να αλλάξει την τιμή του μετρητή προγράμματος.

(δ) Ο φάκελος καταχωρητών γενικού σκοπού παραλαμβάνει από τη κεντρική μονάδα ελέγχου μέχρι 3 διευθύνσεις καταχωρητών για ανάγνωση και εγγραφή, μαζί με τα απαιτούμενα σήματα που καθορίζουν την ή τις μικρολειτουργίες του φακέλου. Δεδομένα για εγγραφή έρχονται αργότερα από την ΑΛΜ ή από τη μνήμη δεδομένων. Τα περιεχόμενα των καταχωρητών που διαβάζονται μπορούν να σταλούν (1) στην ΑΛΜ σαν τελούμενα κάποιας πράξης, (2) στη ΜΔΜ δεδομένων σα δεδομένα για αποθήκευση, είτε (3) στη μονάδα μετρητή προγράμματος σα διεύθυνση ανάγνωσης της επόμενης εντολής.

(ε) Η ΑΛΜ δέχεται μέχρι δύο εισόδους, από τις οποίες η μία έρχεται από το φάκελο καταχωρητών γενικού σκοπού, και η άλλη είτε πάλι από το φάκελο καταχωρητών γενικού σκοπού είτε από τη κεντρική μονάδα ελέγχου σαν άμεσο τελούμενο της εντολής. Με την ολοκλήρωση της πράξης, η ΑΛΜ στέλνει το αποτέλεσμα σε μία από τρεις υπομονάδες της ΜΕΔ: (1) την τιμή του αποτελέσματος στο φάκελο καταχωρητών για εγγραφή, (2) την τιμή του αποτελέσματος στη ΜΔΜ δεδομένων σα διεύθυνση όπου αυτή θα προσπελάσει κάποιο δεδομένο, είτε (3) την έξοδο Zero (μηδενικής τιμής) στη μονάδα μετρητή προγράμματος σα σήμα ελέγχου που επιτρέπει ή όχι την εκτέλεση άλματος με συνθήκη.

(στ) Η μονάδα μνήμης δεδομένων διαβάζει ή αποθηκεύει κάποιο δεδομένο. Στην πρώτη περίπτωση στέλνει το αποτέλεσμα της ανάγνωσης στο φάκελο καταχωρητών γενικού σκοπού.

Στη ΜΕΔ υπάρχουν πολλές υπομονάδες που δέχονται σε μια είσοδο πληροφορία από περισσότερες της μιας πηγές. Θα περιορίσουμε τη μελέτη στη ροή πληροφορίας που περιγράψαμε, όπως προκύπτει από το σχήμα που μελετήσαμε στο μάθημα. Έτσι παρατηρούμε ότι:

1. Η μονάδα PC δέχεται στην είσοδό της (ΣΗΜΕΙΟ 1) τιμή από τις πιο κάτω πηγές:
 - (α) Από τη μονάδα ελέγχου. Η κεντρική μονάδα ελέγχου σε περίπτωση εντολής άμεσου άλματος χωρίς συνθήκη (ή σε περίπτωση εξωτερικής διακοπής που από δω και πέρα θα χειριζόμαστε σαν άμεσο άλμα χωρίς συνθήκη) στέλνει στη μονάδα PC μια καινούργια διεύθυνση την οποία η τελευταία προωθεί στη ΜΔΜ εντολών για ανάγνωση της επόμενης εντολής. Ακόμα, η κεντρική μονάδα ελέγχου σε περίπτωση εντολής άλματος με συνθήκη στέλνει στη μονάδα PC τη μετατόπιση από την τρέχουσα τιμή του μετρητή προγράμματος αυξημένου κατά 1, όπως αυτή δίνεται στην εντολή.
 - (β) Από το φάκελο καταχωρητών γενικού σκοπού. Σε περίπτωση έμμεσου άλματος, ο φάκελος καταχωρητών προωθεί στη μονάδα PC το περιεχόμενο του καταχωρητή που περιέχει τη διεύθυνση στην οποία εκτελείται το άλμα.
 - (γ) Από την ΑΛΜ. Σε περίπτωση άλματος με συνθήκη, η ΑΛΜ αποτιμά τη συνθήκη και στέλνει το λογικό αποτέλεσμα στη μονάδα PC.

Ας σημειωθεί ότι η αύξηση κατά 1 (ή κατά 4 μετρώντας σε επίπεδο byte) και η πρόσθεση της μετατόπισης υλοποιούνται με δύο αθροιστές που δε δείχνονται στο πιο πάνω διάγραμμα για μείωση της πολυπλοκότητάς του, αλλά δείχνονται παρακάτω.
2. Η ΑΛΜ δέχεται τη δεύτερη είσοδό της (ΣΗΜΕΙΟ 2) από τις ακόλουθες πηγές:
 - (α) Από το φάκελο καταχωρητών γενικού σκοπού. Σε εντολές στις οποίες τα τελούμενα είναι διευθύνσεις καταχωρητών, η δεύτερη είσοδος της ΑΛΜ παίρνει τιμή από το φάκελο καταχωρητών.
 - (β) Από την κεντρική μονάδα ελέγχου. Σε εντολές αριθμητικών/λογικών πράξεων με άμεσο τελούμενο, όπως επίσης και σε εντολές προσπέλασης της μνήμης, η δεύτερη είσοδος της ΑΛΜ παίρνει τιμή απ' ευθείας από τη μονάδα ελέγχου της μονάδας επεξεργασίας δεδομένων. Το άμεσο τελούμενο στην πρώτη ή η μετατόπιση της εντολής προσπέλασης της μνήμης στη δεύτερη περίπτωση λαμβάνεται έτσι από τη λέξη εντολής, και συνήθως υφίσταται προέκταση προσήμου – δηλαδή αντιγραφή του προσήμου έως το εύρος μιας λέξης δεδομένων – πριν χρησιμοποιηθεί από την ΑΛΜ.
3. Ο φάκελος καταχωρητών γενικού σκοπού δέχεται την είσοδο δεδομένων εγγραφής (ΣΗΜΕΙΟ 3) από τις ακόλουθες πηγές:
 - (α) Από την ΑΛΜ. Σε εντολές που χρησιμοποιούν την ΑΛΜ εκτός των εντολών άλματος με συνθήκη και των εντολών προσπέλασης στη μνήμη, το αποτέλεσμα της πράξης στέλνεται στο φάκελο καταχωρητών για αποθήκευση.

- (β) Από τη μνήμη δεδομένων. Σε εντολές ανάγνωσης από τη μνήμη, το δεδομένο που επιστρέφει η ΜΔΜ δεδομένων από τη μνήμη στέλνεται στο φάκελο καταχωρητών για αποθήκευση.
- (γ) Από τη μονάδα PC. Σε εντολές άλματος με σύνδεση η διεύθυνση επόμενης εντολής (τιμή μετρητή προγράμματος +1) αποθηκεύεται στο φάκελο καταχωρητών.
4. Ο φάκελος καταχωρητών γενικού σκοπού δέχεται τον αριθμό καταχωρητή εγγραφής (ΣΗΜΕΙΟ 4) από τις ακόλουθες πηγές:
- (α) Από το πεδίο *rt* της λέξης εντολής. Οι εντολές φόρτωσης και άμεσου τελούμενου (εντολές I-type) γράφουν το αποτέλεσμά τους στον καταχωρητή με αριθμό που λαμβάνεται από τα bits 20-16 της λέξης εντολής.
- (β) Από το πεδίο *rd* της λέξης εντολής. Οι εντολές ΑΛΜ κατ' ευθείαν τελούμενων, ολίσθησης και έμμεσου άλματος (εντολές R-type) γράφουν το αποτέλεσμά τους στον καταχωρητή με αριθμό που λαμβάνεται από τα bits 15-11 της λέξης εντολής.
- (γ) Από τη σταθερά 31. Η εντολή άμεσου άλματος με σύνδεση *jal* γράφει πάντα στο υπονοούμενο τελούμενο καταχωρητή \$31 (ή συμβολικά \$ra).

Σε όλες τις παραπάνω περιπτώσεις, η επιλογή μεταξύ των εναλλακτικών εισόδων γίνεται με τη βοήθεια πολυπλεκτών. Στην πρώτη περίπτωση που έχουμε επιλογή από 3 πηγές – οι οποίες γίνονται 4 αν θεωρήσουμε τον αθροιστή που αυξάνει την τιμή του μετρητή προγράμματος κατά 1 σαν τέταρτη πηγή, χρειαζόμαστε έναν πολυπλέκτη 4×1. Στη δεύτερη περίπτωση που έχουμε επιλογή από 3 πηγές – συμπεριλαμβανοντας και την επιλογή προέκτασης προσήμου, χρειαζόμαστε έναν πολυπλέκτη 4×1. Στην τρίτη περίπτωση μπορούμε να χρησιμοποιήσουμε είτε έναν πολυπλέκτη 4×1 είτε δύο διαδοχικούς πολυπλέκτες 2×1 για την πρώτη, και έναν πολυπλέκτη 2×1 για τη δεύτερη επιλογή. Στην τέταρτη περίπτωση έχουμε 3 πηγές και άρα χρειαζόμαστε πάλι πολυπλέκτη 4×1. Πιο αναλυτικά:

1. Στη μονάδα PC ο μετρητής προγράμματος αυξάνεται σε κάθε κύκλο εντολής κατά 1. Επομένως, σε περίπτωση άλματος θα πρέπει να αγνοήσουμε την αυξημένη κατά 1 τιμή και να θέσουμε στο μετρητή προγράμματος την τιμή που υπαγορεύει το άλμα. Έτσι η κεντρική μονάδα ελέγχου απαιτείται να στείλει στη μονάδα PC ένα σήμα ελέγχου – έστω X – που να έχει τιμή 1 όταν έχουμε εντολή άλματος και τιμή 0 διαφορετικά.

Επειδή όμως έχουμε περισσότερες από μία διαφορετικές εντολές άλματος, ένα μοναδικό σήμα από την κεντρική μονάδα ελέγχου δεν είναι αρκετό για την επιλογή νέας τιμής στο μετρητή προγράμματος. Πέρα από το παραπάνω σήμα, θα θεωρήσουμε ότι η μονάδα ελέγχου με την αποκωδικοποίηση της εντολής στέλνει στη μονάδα PC δύο ακόμα σήματα ελέγχου, ένα σήμα – έστω Y – που έχει τιμή 1 όταν το άλμα είναι έμμεσο, και 0 όταν αυτό είναι άμεσο, κι ένα σήμα – έστω W – που έχει τιμή 1 όταν η διευθυνσιοδότηση του άλματος είναι σχετική και 0 όταν αυτή είναι απόλυτη.

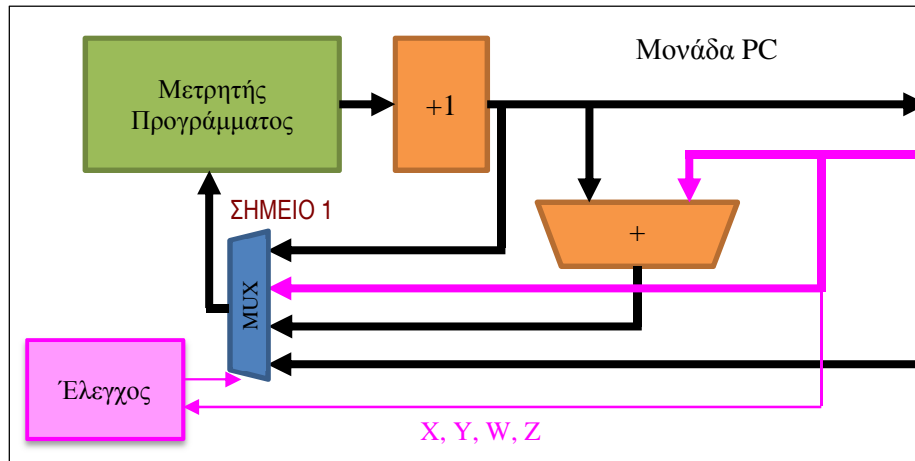
Το σήμα Y μας βοηθά να επιλέξουμε αν η νέα τιμή του μετρητή προγράμματος θα ληφθεί από το φάκελο καταχωρητών ή απ' ευθείας από τη μονάδα ελέγχου. Το σήμα W μας βοηθά να επιλέξουμε αν η τιμή που παίρνουμε προστίθεται στην αυξημένη κατά 1 τιμή του μετρητή προγράμματος ή αν αυτή εισάγεται απ' ευθείας σ' αυτόν.

Παρά τα σήματα X, Y και W που λαμβάνουμε από τη μονάδα ελέγχου, σε περίπτωση άλματος με συνθήκη, με τη συνθήκη να προκύπτει ψευδής, το άλμα δεν εκτελείται, παρόλο που η εντολή είναι εντολή άλματος. Συμπεραίνουμε έτσι ότι τα παραπάνω σήματα δεν αρκούν για την επιλογή νέας τιμής στο μετρητή προγράμματος. Όπως θα περιμέναμε, χρησιμοποιούμε την έξοδο μηδενικής τιμής (Zero) της ΑΛΜ σαν ένα ακόμα σήμα ελέγχου Z που λαμβάνει τιμή από το σήμα Zero, ώστε να επιτρέψουμε στη μονάδα PC να αλλάξει τιμή στο μετρητή προγράμματος μόνο όταν η συνθήκη άλματος είναι αληθής.

Έχοντας δει μέχρι τώρα τις δυνατές πηγές νέας τιμής μετρητή προγράμματος και τα σήματα ελέγχου που χρειαζόμαστε για την επιλογή μιας από αυτών, μπορούμε να προχωρήσουμε στην εύρεση των σημάτων ελέγχου του πολυπλέκτη 4×1 που χρησιμοποιούμε γι' αυτή την επιλογή. Κωδικοποιούμε τις δυνατές τιμές που μπορεί να πάρει ο μετρητής προγράμματος ως εξής:

Κωδικός	Επιλογή
00	PC = PC + 1
01	PC = απόλυτη διεύθυνση ¹
10	PC = PC + 1 + σχετική διεύθυνση
11	PC = τιμή καταχωρητή

Η μονάδα PC θα έχει έτσι την αναλυτική μορφή του παρακάτω διαγράμματος:



όπου οι γραμμές σύνδεσης με την υπόλοιπη ΜΕΔ είναι ταυτόσημες με αυτές του προηγούμενου διαγράμματος.

Μπορούμε να ολοκληρώσουμε τη μελέτη της περίπτωσης αυτής με την εύρεση του πίνακα αλήθειας των ψηφίων του κωδικού επιλογής σε συνάρτηση των τιμών των σημάτων X, Y, W και Z²:

X	Y	W	Z	MUX1	MUX0
0	x	x	x	0	0
1	0	0	x	0	1
1	0	1	0	0	0
1	0	1	1	1	0
1	1	x	x	1	1

απ' όπου παίρνουμε τις λογικές συναρτήσεις που υλοποιούν τον έλεγχο του πολυπλέκτη:

$$\text{MUX1} = X * (Y + W * Z)$$

$$\text{MUX0} = X * (Y + W')$$

Ο παραπάνω σχεδιασμός της διαδικασίας επιλογής της σωστής τιμής εισόδου της μονάδας του μετρητή προγράμματος δεν είναι μοναδικός. Πέρα από την επιλογή του κυκλώματος πολυπλέκτη για την τελική επιλογή, η επιλογή των σημάτων ελέγχου, καθώς και η πιο πάνω κωδικοποίηση των ψηφίων επιλογής, δε συνιστούν τη μοναδική λύση του προβλήματος. Όπως είδαμε και στην προηγούμενη άσκηση, η υλοποίηση μονάδων ελέγχου είναι τόσο περίπλοκη διαδικασία, που δε μπορεί να έχει ούτε μονοσήμαντη ούτε αποδεδειγμένα βέλτιστη λύση.

¹ Στην πραγματικότητα το άμεσο άλμα δεν υλοποιείται με πλήρως απόλυτη διευθυνσιοδότηση – δηλαδή λήψη της πλήρους διεύθυνσης από τη μονάδα ελέγχου, αλλά με παράθεση των 26 λιγότερο σημαντικών ψηφίων της λέξης εντολής με τα 4 πιο σημαντικά ψηφία του αυξημένου PC. Αν ο PC περιέχει διεύθυνση επιπέδου byte, τότε τα 2 λιγότερο σημαντικά ψηφία του τίθενται αυτόματα σε “00”.

² Εδώ δείχνουμε την υλοποίηση για την εντολή διακλάδωσης beq μόνο. Αν θέλουμε να επεκτείνουμε την υλοποίηση και για την εντολή bne, θα πρέπει να τροποποιήσουμε κατάλληλα το σήμα Z, ώστε να συνδυάζει το σήμα Zero της ALM με τον κωδικό λειτουργίας της εντολής.

2. Στην περίπτωση της επιλογής της δεύτερης εισόδου στην ΑΛΜ η σχεδίαση είναι απλούστερη. Για τρεις πηγές, η επιλογή γίνεται με έναν πολυπλέκτη 4×1, ο οποίος απαιτεί δύο bits ελέγχου από την κεντρική μονάδα ελέγχου. Όπως προκύπτει από μελέτη των εντολών που χρησιμοποιούν την ΑΛΜ, τα δύο αυτά bits μπορούμε να τα συνδέσουμε απ' ευθείας σε δύο ανεξάρτητα σήματα ελέγχου, χωρίς να χρειαστεί να κωδικοποιήσουμε τις επιλογές.

Έτσι, μπορούμε να αντιστοιχίσουμε το ένα από τα δύο bits ελέγχου στο σήμα επιλογής με τιμή 1 αν η ΑΛΜ δέχεται τη δεύτερη είσοδο απ' ευθείας από τη μονάδα ελέγχου και ειδικότερα από τα 16 λιγότερο σημαντικά ψηφία της λέξης εντολής, και με τιμή 0 αν η ΑΛΜ δέχεται τη δεύτερη είσοδο από το φάκελο καταχωρητών γενικού σκοπού. Το δεύτερο bit ελέγχου μπορούμε να το αντιστοιχίσουμε στο σήμα επιλογής με τιμή 1 αν η σταθερά που λαμβάνεται από τη λέξη εντολής υφίσταται προέκταση προσήμου, οπότε το πιο σημαντικό από τα 16 bits αντιγράφεται μέχρι το εύρος των 32 bits της ΑΛΜ, και με τιμή 0 αν η σταθερά προεκτείνεται με μηδενικά ψηφία.

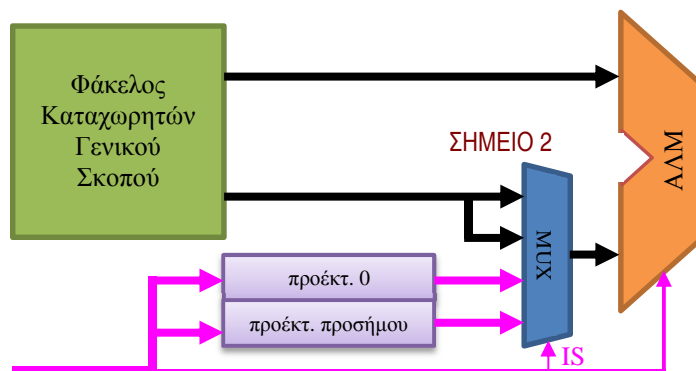
Έστω I το πρώτο και S το δεύτερο σήμα. Εντολές όπως προσπελάσεις μνήμης και εντολές ΑΛΜ με άμεσο τελούμενο απαιτούν τιμή 1, ενώ εντολές όπως διακλαδώσεις και εντολές ΑΛΜ με κατ' ευθείαν τελούμενα απαιτούν τιμή 0 για το σήμα I. Από τις πρώτες, οι προσπελάσεις μνήμης και οι αριθμητικές εντολές απαιτούν 1, ενώ οι λογικές εντολές απαιτούν 0 για το σήμα S. Για τις δεύτερες, το σήμα S είναι αδιάφορο. Αν συνδέσουμε το πρώτο σήμα στο πιο σημαντικό και το δεύτερο σήμα στο λιγότερο σημαντικό ψηφίο επιλογής του πολυπλέκτη, τότε προκύπτει ο παρακάτω πίνακας επιλογών:

Κωδικός	Επιλογή
00	ΑΛΜ2 = ΦΚ
01	ΑΛΜ2 = ΦΚ
10	ΑΛΜ2 = σταθερά χωρίς προέκταση προσήμου
11	ΑΛΜ2 = σταθερά με προέκταση προσήμου

Παρατηρήστε ότι συνδέουμε την έξοδο του ΦΚ και στις δύο επιλογές 00 και 01, επειδή το σήμα S είναι αδιάφορο όταν I = 0. Επομένως, οι λογικές συναρτήσεις ελέγχου του πολυπλέκτη είναι απλά όπως τις θέλουμε:

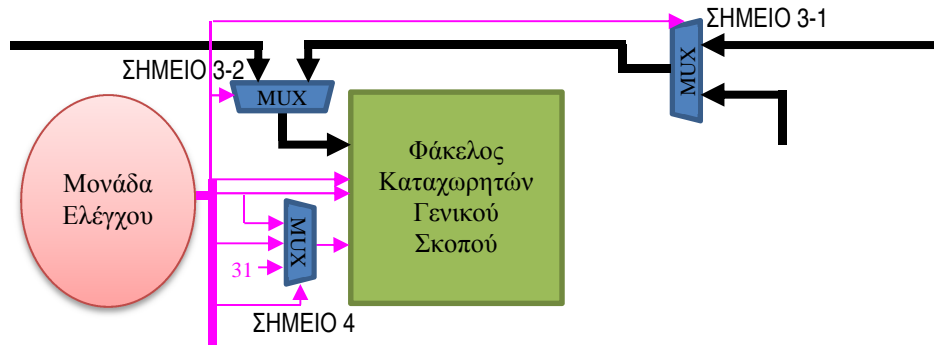
$$\begin{aligned} \text{MUX1} &= I \\ \text{MUX0} &= S \end{aligned}$$

Μπορούμε τέλος να σχεδιάσουμε και την αναλυτική σύνδεση της δεύτερης εισόδου της ΑΛΜ:



3. Η επιλογή τιμής στην είσοδο δεδομένων εγγραφής του φακέλου καταχωρητών μπορεί να γίνει με δύο διαδοχικούς πολυπλέκτες 2×1. Επιλέγουμε αυτή τη λύση επειδή η επιλογή μεταξύ εισόδου από την ΑΛΜ και εισόδου από τη μνήμη δεδομένων μπορεί να γίνει κοντά στις δύο αυτές υπομονάδες, ενώ η τελική επιλογή μπορεί να γίνει κοντά στην είσοδο του φακέλου καταχωρητών. Έτσι, η διαδικασία επιλογής γίνεται απλή. Το σήμα ελέγχου που θα στέλνει η κεντρική μονάδα ελέγχου για τον πρώτο πολυπλέκτη έχει τιμή 1 όταν η είσοδος γίνεται από

τη μνήμη δεδομένων και 0 όταν η είσοδος γίνεται από την ΑΛΜ (ΣΗΜΕΙΟ 3-1). Αντίστοιχα, το σήμα της μονάδας ελέγχου για το δεύτερο πολυπλέκτη θα έχει τιμή 1 όταν η είσοδος γίνεται από τη μονάδα PC και 0 όταν η είσοδος έρχεται από την έξοδο του πρώτου πολυπλέκτη (ΣΗΜΕΙΟ 3-2). Η αναλυτική διάταξη των δύο πολυπλεκτών, μαζί με τις γραμμές σημάτων επιλογής που δε φαίνονται στο αρχικό διάγραμμα, θα είναι:



4. Η δεύτερη επιλογή στο φάκελο καταχωρητών αφορά την επιλογή του σωστού αριθμού καταχωρητή εγγραφής αποτελέσματος είτε από bits της λέξης εντολής είτε από την καλωδιωμένη σταθερά 31. Εδώ έχουμε και πάλι τρεις επιλογές, και θα χρησιμοποιήσουμε έναν πολυπλέκτη 4×1. Η επιλογή με τιμή ελέγχου 00 αντιστοιχεί στα bits 20-16 (πεδίο rt) της λέξης εντολής, η επιλογή με τιμή ελέγχου 01 αντιστοιχεί στα bits 15-11 (πεδίο rd) της λέξης εντολής, η επιλογή με τιμή ελέγχου 10 αντιστοιχεί στη σταθερά 31, ενώ η τελευταία επιλογή του πολυπλέκτη μένει ασύνδετη. Εναλλακτικά, όπως και παραπάνω, μπορούμε να συνδέσουμε τη σταθερά 31 και στην τέταρτη επιλογή, ώστε να χρησιμοποιήσουμε ως σημαντικότερο bit ελέγχου επιλογής το σήμα άλματος με σύνδεση, και ως λιγότερο σημαντικό bit ελέγχου επιλογής το σήμα επιλογής μεταξύ των δύο πεδίων της λέξης εντολής. Το αναλυτικό διάγραμμα σύνδεσης του πολυπλέκτη είναι ενσωματωμένο στο προηγούμενο.

Για όλες τις πιο πάνω επιλογές, το διάγραμμα που μελετήσαμε στο μάθημα έχει ονομάσει συμβολικά τους πολυπλέκτες της ΜΕΔ. Έτσι, ο πολυπλέκτης που αφορά την επιλογή στην είσοδο του PC (ΣΗΜΕΙΟ 1) έχει ονομαστεί PCSrc, ο πολυπλέκτης της δεύτερης εισόδου της ΑΛΜ (ΣΗΜΕΙΟ 2) έχει ονομαστεί ALUSrcB, ο πρώτος πολυπλέκτης που καθορίζει την πηγή των δεδομένων εγγραφής του φακέλου καταχωρητών έχει ονομαστεί MemtoReg (ΣΗΜΕΙΟ 3-1), ο δεύτερος πολυπλέκτης για την ίδια πηγή έχει ονομαστεί PCtoReg (ΣΗΜΕΙΟ 3-2), και τέλος ο πολυπλέκτης αριθμού καταχωρητή εγγραφής έχει ονομαστεί RegDst (ΣΗΜΕΙΟ 4). Με το ίδιο όνομα αναφερόμαστε και στα σήματα που οδηγούν τους πολυπλέκτες. Όταν ένας πολυπλέκτης έχει περισσότερες από δύο επιλογές, το σήμα του αποτελείται από πολλαπλά ψηφία. Τότε, τα επιμέρους ψηφία διαχωρίζονται με δείκτες. Έτσι, για τον PCSrc – όπου έχουμε τέσσερις επιλογές, το αντίστοιχο σήμα έχει δύο ψηφία, τα PCSrc₁ ≡ MUX1 και PCSrc₀ ≡ MUX0.

Άσκηση 3:

Θεωρήστε τη μονάδα επεξεργασίας δεδομένων της απλοποιημένης αρχιτεκτονικής MIPS, στην οποία κάθε εντολή ολοκληρώνεται σε έναν κύκλο μηχανής. Δείξτε πώς εκτελείται στην αρχιτεκτονική αυτή το παρακάτω υποπρόγραμμα `zero(int a[], int n)`:

```

zero: ori    $8, $0, 0
loop: sll    $9, $8, 2
      add    $9, $9, $4
      sw     $0, 0($9)
      sw     $0, 4($9)
      addi   $8, $8, 2

```

```

slt    $9, $8, $5
bne   $9, $0, loop
jr    $31

```

Απάντηση:

Το υποπρόγραμμα που μας δίνεται μηδενίζει τα n στοιχεία ενός μονοδιάστατου πίνακα ακεραίων a . Παρατηρούμε ότι ο μηδενισμός γίνεται ανά δύο στοιχεία σε κάθε επανάληψη του βρόχου `loop`, κάτι που σημαίνει ότι το n πρέπει να είναι άρτιος αριθμός, αλλιώς ενδέχεται να σημειωθεί σφάλμα στην εκτέλεση του κώδικα.

Ξαναγράφουμε τον κώδικα του υποπρογράμματος `zero()` με τη μετάφραση κάθε εντολής σε γλώσσα μηχανής:

zero:	ori	\$8, \$0, 0	001101 00000 01000 0000000000000000
loop:	sll	\$9, \$8, 2	000000 00000 01000 01001 00010 000000
	add	\$9, \$9, \$4	000000 01001 00100 01001 00000 100000
	sw	\$0, 0(\$9)	101011 01001 00000 0000000000000000
	sw	\$0, 4(\$9)	101011 01001 00000 0000000000000100
	addi	\$8, \$8, 2	001000 01000 01000 0000000000000010
	slt	\$9, \$8, \$5	000000 01000 00101 01001 00000 101010
	bne	\$9, \$0, loop	000101 01001 00000 111111111111001
	jr	\$31	000000 11111 00000 00000 00000 001000

Παρατηρούμε ότι οι μισές περίπου από τις εντολές έχουν άμεσο τελούμενο. Ειδικά η εντολή ολίσθησης (η δεύτερη εντολή στον κώδικα) έχει ένα άμεσο τελούμενο μήκους 5 bits. Επειδή ολίσθηση για περισσότερα από 31 bits δεν υποστηρίζεται με αυτή την εντολή, η κωδικοποίηση της εντολής γίνεται όπως η κωδικοποίηση εντολών χωρίς άμεσο τελούμενο, με την ομάδα των 5 bits που καθορίζουν τον αριθμό ψηφίων ολίσθησης να βρίσκεται στην τελευταία προς τα δεξιά ομάδα των 5 bits, αμέσως δηλαδή πριν τον κωδικό τελεστή. Η μονάδα ελέγχου θα αναγνωρίσει την ομάδα αυτή των 5 bits σαν άμεσο τελούμενο και θα τη στείλει απ' ευθείας στην ΑΛΜ, με τον ίδιο τρόπο που στέλνει σ' αυτήν τα άμεσα τελούμενα των 16 bits³.

Με την έναρξη εκτέλεσης του υποπρογράμματος `zero()` η μονάδα μετρητή προγράμματος στέλνει στη ΜΔΜ εντολών τη διεύθυνση που αντιστοιχεί στην πρώτη εντολή του υποπρογράμματος, ενώ παράλληλα αυξάνει το μετρητή προγράμματος κατά 1. Η μνήμη εντολών ανακαλεί από τη μνήμη τη λέξη 00110100000010000000000000000000

Η μονάδα ελέγχου αποκωδικοποιεί την εντολή, αναγνωρίζοντας στον κωδικό λειτουργίας 001101 την εντολή `ori` (λογικό Η με άμεσο τελούμενο). Στη συνέχεια η μονάδα ελέγχου στέλνει στην ΑΛΜ τον κωδικό που αντιστοιχεί στη μικρολειτουργία του λογικού Η⁴, τα 16 bits της σταθεράς 0000000000000000 και ένα σήμα ελέγχου που κατευθύνει την ΑΛΜ να δεχτεί τη δεύτερη είσοδό της απ' ευθείας από τη μονάδα ελέγχου, με προέκταση μηδενικών. Επίσης στέλνει στο φάκελο καταχωρητών γενικού σκοπού τις ομάδες των 5 bits 00000 και 01000, από τις οποίες η πρώτη αντιστοιχεί σε διεύθυνση ανάγνωσης και η δεύτερη σε διεύθυνση εγγραφής, καθώς και ένα σήμα ελέγχου που καθορίζει ότι η είσοδος εγγραφής του φακέλου θα γίνει από την ΑΛΜ και όχι από τη ΜΔΜ δεδομένων. Τέλος η μονάδα ελέγχου στέλνει στη μονάδα PC έναν κωδικό ελέγχου που αντιστοιχεί σε εντολές χωρίς άλμα, και στη ΜΔΜ δεδομένων έναν κωδικό ελέγχου που αντιστοιχεί σε εντολές χωρίς προσπέλαση της μνήμης δεδομένων.

³ Η μικρολειτουργία της ολίσθησης συνήθως εκτελείται σε ανεξάρτητη υπομονάδα, και όχι στην ΑΛΜ. Εδώ για απλούστευση θεωρούμε ότι εκτελείται στην ΑΛΜ.

⁴ Ο κωδικός της μικρολειτουργίας στην πραγματικότητα παράγεται από την υπομονάδα ελέγχου της ΑΛΜ με συνδυασμό των κωδικών λειτουργίας και τελεστή της εντολής. Η υπομονάδα αυτή δεν είναι απαραίτητο να βρίσκεται κοντά στην ΑΛΜ, αλλά μπορεί να είναι υπομονάδα της κεντρικής μονάδας ελέγχου.

Ο φάκελος καταχωρητών διαβάζει το περιεχόμενο του καταχωρητή \$0 (που στην αρχιτεκτονική MIPS είναι πάντα 0) και το προωθεί στην ΑΛΜ. Επίσης αναμένει στην είσοδο εγγραφής τη λέξη που θα εγγράψει στον καταχωρητή \$8, η οποία θα περάσει στο φάκελο με την πτώση του παλμού του κύκλου μηχανής. Στην πραγματικότητα ο φάκελος καταχωρητών στέλνει ό,τι υπάρχει στη δεύτερη πόρτα ανάγνωσης προς την ΑΛΜ και προς τη ΜΔΜ δεδομένων. Όμως, η μεν πρώτη ροή πληροφορίας καταλήγει σε έναν πολυπλέκτη που δεν την επιλέγει σαν είσοδο της ΑΛΜ, η δε δεύτερη καταλήγει μεν στη ΜΔΜ δεδομένων, αλλά όχι στη μνήμη δεδομένων, εφ' όσον η μονάδα ελέγχου έχει ενημερώσει τη ΜΔΜ δεδομένων ότι δεν έχουμε εντολή αποθήκευσης στη μνήμη.

Η ΑΛΜ εκτελεί τη μικρολειτουργία του λογικού Η μεταξύ του περιεχομένου του καταχωρητή \$0 και της σταθεράς 0, και στέλνει το αποτέλεσμα της, που είναι η τιμή 0, στο φάκελο καταχωρητών. Το αποτέλεσμα της πράξης στέλνεται επίσης σε διεύθυνση στη ΜΔΜ δεδομένων, όπου δε χρησιμοποιείται λόγω του κωδικού που η τελευταία έχει παραλάβει από τη μονάδα ελέγχου, και σε λογικό αποτέλεσμα στη μονάδα PC, όπου επίσης δε χρησιμοποιείται, αφού η τελευταία γνωρίζει ήδη ότι δεν έχουμε εντολή άλματος.

Η ΜΔΜ δεδομένων δεν εκτελεί καμία λειτουργία σε αυτή την εντολή, ενώ η όποια τιμή που υπάρχει στην έξοδό της δεν επιλέγεται για αποθήκευση στο φάκελο καταχωρητών.

Για τις επόμενες εντολές η φάση ανάκλησης επαναλαμβάνεται και δε θα ξαναπεριγραφτεί. Η μονάδα ελέγχου, και για κάθε εντολή του συγκεκριμένου υποπρογράμματος zero(), στέλνει στο φάκελο καταχωρητών το σήμα που καθορίζει ότι η είσοδος εγγραφής – όταν έχουμε εγγραφή – γίνεται από την ΑΛΜ, μια που στον κώδικα που μας δίνεται δεν έχουμε εντολή ανάγνωσης από τη μνήμη δεδομένων. Δε θα ξανααναφέρουμε την αποστολή του σήματος αυτού, ενώ δε θα αναφερόμαστε ούτε στον κωδικό που η μονάδα ελέγχου στέλνει στη ΜΔΜ δεδομένων για εντολές χωρίς προσπέλαση στη μνήμη, ούτε στον κωδικό που η ίδια μονάδα στέλνει στη μονάδα PC για εντολές χωρίς άλμα. Τέλος, για περαιτέρω απλούστευση της περιγραφής δε θα ξανααναφέρουμε μετάδοση πληροφορίας που δε χρησιμοποιείται, ούτε θα αναφερόμαστε σε υπομονάδες που δεν εκτελούν κάποια λειτουργία σε μια συγκεκριμένη εντολή.

Αφού λοιπόν ανακληθεί η δεύτερη εντολή, η μονάδα ελέγχου την αποκωδικοποιεί, αναγνωρίζοντας στο συνδυασμό κωδικού λειτουργίας και κωδικού τελεστή την εντολή sll (αριστερή ολίσθηση με άμεσο τελούμενο). Στη συνέχεια η μονάδα ελέγχου στέλνει στην ΑΛΜ τον κωδικό που αντιστοιχεί στη μικρολειτουργία της αριστερής ολίσθησης, τα 5 bits της σταθεράς 2 που δηλώνει τον αριθμό ψηφίων ολίσθησης και ένα σήμα ελέγχου που κατευθύνει την ΑΛΜ να δεχτεί τη δεύτερη είσοδό της απ' ευθείας από τη μονάδα ελέγχου. Επίσης στέλνει στο φάκελο καταχωρητών γενικού σκοπού τις ομάδες των 5 bits 01000 (διεύθυνση ανάγνωσης) και 01001 (διεύθυνση εγγραφής).

Ο φάκελος καταχωρητών διαβάζει το περιεχόμενο του καταχωρητή \$8 και το προωθεί στην ΑΛΜ. Επίσης αναμένει στην είσοδο εγγραφής τη λέξη που θα εγγράψει στον καταχωρητή \$9.

Η ΑΛΜ εκτελεί τη μικρολειτουργία της αριστερής ολίσθησης του περιεχομένου του καταχωρητή \$8 κατά 2 ψηφία, και στέλνει το αποτέλεσμά της στο φάκελο καταχωρητών.

Όσο αφορά την τρίτη εντολή, η μονάδα ελέγχου την αποκωδικοποιεί, αναγνωρίζοντας στο συνδυασμό κωδικού λειτουργίας και κωδικού τελεστή την εντολή add (πρόσθεση με κατ' ευθείαν τελούμενα). Στη συνέχεια στέλνει στην ΑΛΜ τον κωδικό που αντιστοιχεί στη μικρολειτουργία της πρόσθεσης και ένα σήμα ελέγχου που κατευθύνει την ΑΛΜ να δεχτεί τη δεύτερη είσοδό της από το φάκελο καταχωρητών. Επίσης στέλνει στο φάκελο καταχωρητών γενικού σκοπού τις ομάδες των 5 bits 01001 (διεύθυνση ανάγνωσης), 00100 (διεύθυνση ανάγνωσης) και 01001 (διεύθυνση εγγραφής).

Ο φάκελος καταχωρητών διαβάζει τα περιεχόμενα των καταχωρητών \$9 και \$4 και τα προωθεί στην ΑΛΜ. Επίσης αναμένει στην είσοδο εγγραφής τη λέξη που θα εγγράψει στον καταχωρητή \$9.

Η ΑΛΜ εκτελεί τη μικρολειτουργία της πρόσθεσης των περιεχομένων των καταχωρητών \$9 και \$4, και στέλνει το αποτέλεσμά της στο φάκελο καταχωρητών.

Η τέταρτη εντολή είναι εντολή αποθήκευσης στη μνήμη δεδομένων. Η μονάδα ελέγχου την αποκωδικοποιεί, αναγνωρίζοντας στον κωδικό λειτουργίας την εντολή sw (αποθήκευση λέξης). Στη συνέχεια στέλνει στην ΑΛΜ τον κωδικό που αντιστοιχεί στη μικρολειτουργία της πρόσθεσης, τα 16 bits της μετατόπισης 0000000000000000 και ένα σήμα ελέγχου που κατευθύνει την ΑΛΜ να δεχτεί τη δεύτερη είσοδό της απ' ευθείας από τη μονάδα ελέγχου, με προέκταση προσήμου. Επίσης στέλνει στο φάκελο καταχωρητών γενικού σκοπού τις ομάδες των 5 bits 01001 (διεύθυνση ανάγνωσης) και 00000 (διεύθυνση ανάγνωσης), κι ένα σήμα ελέγχου που αποτρέπει εγγραφή στο φάκελο. Τέλος στέλνει στη ΜΔΜ δεδομένων τον κωδικό ελέγχου για εντολές αποθήκευσης στη μνήμη δεδομένων.

Ο φάκελος καταχωρητών διαβάζει τα περιεχόμενα των καταχωρητών \$9 και \$0 και προωθεί το μεν πρώτο στην ΑΛΜ, το δε δεύτερο στη ΜΔΜ δεδομένων, ενώ δεν αναμένει στην είσοδο εγγραφής κανένα δεδομένο για εγγραφή.

Η ΑΛΜ εκτελεί τη μικρολειτουργία της πρόσθεσης του περιεχομένου του καταχωρητή \$9 με τη σταθερά μετατόπισης 0, και στέλνει το αποτέλεσμα της στη ΜΔΜ δεδομένων σα διεύθυνση προσπέλασης στη μνήμη.

Η ΜΔΜ δεδομένων στέλνει το περιεχόμενο του καταχωρητή \$0 στη μνήμη δεδομένων.

Η επόμενη εντολή είναι επίσης εντολή αποθήκευσης στη μνήμη δεδομένων. Η προηγούμενη περιγραφή ισχύει και γι' αυτή την εντολή, με μόνη διαφορά ότι τα 16 bits της μετατόπισης που στέλνονται από τη μονάδα ελέγχου στην ΑΛΜ είναι τώρα 0000000000000100. Η ΑΛΜ με τη σειρά της προσθέτει τη νέα σταθερά μετατόπισης 4 στο περιεχόμενο του καταχωρητή \$9, και στέλνει το αποτέλεσμα της στη ΜΔΜ δεδομένων σα διεύθυνση προσπέλασης στη μνήμη. Παρατηρούμε ότι η νέα διεύθυνση εγγραφής αντιστοιχεί στη λέξη που ακολουθεί αυτήν που εγγράφηκε με την προηγούμενη εντολή. Αν ο αριθμός n των στοιχείων του πίνακα δεν είναι άρτιος, στην τελευταία επανάληψη του βρόχου η εντολή αυτή θα αποθηκεύσει σε χώρο εκτός του δεσμευμένου χώρου μνήμης του πίνακα, κάτι που είναι πιθανό να οδηγήσει σε σφάλμα εκτέλεσης.

Η έκτη εντολή εκτελείται όπως η πρώτη και η επόμενη εκτελείται όπως η τρίτη. Για την πρώτη από τις δύο αυτές εντολές, η μονάδα ελέγχου αναγνωρίζει την εντολή addi (πρόσθεση με άμεσο τελούμενο). Τα σήματα ελέγχου είναι τα αντίστοιχα, ενώ η διεύθυνση καταχωρητή ανάγνωσης είναι τώρα 01000, η διεύθυνση καταχωρητή εγγραφής είναι 01000, ενώ η σταθερά που αποστέλλεται στην ΑΛΜ είναι η 0000000000000010, η οποία υφίσταται προέκταση προσήμου. Στην έβδομη εντολή η μονάδα ελέγχου αναγνωρίζει την εντολή slt (σύγκριση μικρότερου με κατ' ευθείαν τελούμενα). Οι καταχωρητές ανάγνωσης είναι τώρα οι \$8 και \$5, ενώ ο καταχωρητής εγγραφής είναι ο \$9.

Η όγδοη εντολή είναι εντολή άλματος με συνθήκη. Η μονάδα ελέγχου που αποκωδικοποιεί την εντολή, αναγνωρίζει στον κωδικό λειτουργίας 000101 την εντολή bne (άλμα με συνθήκη ανισότητας). Στη συνέχεια η μονάδα ελέγχου στέλνει στην ΑΛΜ τον κωδικό που αντιστοιχεί στη μικρολειτουργία της σύγκρισης για ανισότητα⁵, και ένα σήμα ελέγχου που κατευθύνει την ΑΛΜ να δεχτεί τη δεύτερη είσοδό της από το φάκελο καταχωρητών. Επίσης στέλνει στο φάκελο καταχωρητών γενικού σκοπού τις ομάδες των 5 bits 01001 και 00000, οι οποίες αντιστοιχούν σε διευθύνσεις ανάγνωσης, και το σήμα ελέγχου που αποτρέπει εγγραφή στο φάκελο. Τέλος η μονάδα ελέγχου στέλνει στη μονάδα PC τον κωδικό ελέγχου που αντιστοιχεί σε εντολές άλματος με συνθήκη και τα 16 bits της σταθεράς 111111111111001.

Ο φάκελος καταχωρητών διαβάζει το περιεχόμενο των καταχωρητών \$9 και \$0 και τα προωθεί στην ΑΛΜ, ενώ δεν αναμένει δεδομένο εγγραφής.

⁵ Η πράξη της σύγκρισης για ανισότητα μπορεί να γίνει εύκολα με τη μικρολειτουργία της αφαίρεσης, οπότε η λογική τιμή του αποτελέσματος λαμβάνεται με αντιστροφή του σήματος μηδενικής εξόδου Z της ΑΛΜ. Και οι δύο εντολές διακλάδωσης beq και bne μπορούν να υποστηριχθούν ταυτόχρονα με πράξη αποκλειστικού Η μεταξύ του σήματος Z και κατάλληλου σήματος ελέγχου που τις διαχωρίζει.

Η ΑΛΜ εκτελεί τη μικρολειτουργία της σύγκρισης για ανισότητα μεταξύ του περιεχομένου του καταχωρητή \$9 και του περιεχομένου του καταχωρητή \$0 (άσχετα αν το τελευταίο είναι πάντα 0), και στέλνει το λογικό αποτέλεσμα της (δηλαδή ένα σήμα 1 bit) στη μονάδα μετρητή προγράμματος.

Η μονάδα PC τώρα προσθέτει τη σταθερά -7 που παρέλαβε από τη μονάδα ελέγχου στην αυξημένη κατά 1 τιμή του μετρητή προγράμματος, και αναγνωρίζοντας τόσο τον κωδικό ελέγχου για άλμα με συνθήκη, όσο και το σήμα που παρέλαβε από την ΑΛΜ, είτε προωθεί το αποτέλεσμα της πρόσθεσης στο μετρητή προγράμματος, εάν το τελευταίο σήμα είχε τη λογική τιμή 1, είτε διατηρεί σε αυτόν την ήδη αυξημένη κατά 1 προηγούμενη τιμή του, εάν το σήμα αυτό είχε τη λογική τιμή 0.

Με την ολοκλήρωση της εκτέλεσης της παραπάνω εντολής άλματος με συνθήκη, είτε επιστρέφουμε στη θέση που στο υποπρόγραμμα zero() αντιστοιχεί στην ετικέτα loop, είτε συνεχίζουμε με την επόμενη εντολή. Μπορούμε δε εύκολα να διαπιστώσουμε ότι ο βρόχος που δημιουργείται εκτελεί συνολικά $n/2$ επαναλήψεις. Κάθε φορά η σειρά και ο τρόπος εκτέλεσης των εντολών είναι τα ίδια με όσα περιγράψαμε μέχρι τώρα.

Με την έξοδο από το βρόχο, η τελευταία τιμή αποτελέσματος της ΑΛΜ είναι το λογικό 0. Έτσι η μονάδα PC δεν εκτελεί το άλμα, και η ροή εντολών συνεχίζεται με την επόμενη και τελευταία εντολή του υποπρόγραμματος. Αυτή αποκωδικοποιείται από τη μονάδα ελέγχου που αναγνωρίζει στον κωδικό τελεστή 001000 την εντολή jr (έμμεσο άλμα χωρίς συνθήκη). Η μονάδα ελέγχου στέλνει στο φάκελο καταχωρητών γενικού σκοπού την ομάδα των 5 bits 1111 (διεύθυνση ανάγνωσης) και το σήμα ελέγχου που αποτρέπει εγγραφή στο φάκελο. Ακόμα η μονάδα ελέγχου στέλνει στη μονάδα PC τον κωδικό ελέγχου που αντιστοιχεί σε έμμεσες εντολές άλματος χωρίς συνθήκη.

Ο φάκελος καταχωρητών διαβάζει το περιεχόμενο του καταχωρητή \$31 και το προωθεί στη μονάδα PC, ενώ δεν αναμένει δεδομένο εγγραφής.

Η μονάδα PC τέλος, αναγνωρίζοντας τον κωδικό ελέγχου για έμμεσο άλμα χωρίς συνθήκη, προωθεί το δεδομένο που παρέλαβε από το φάκελο καταχωρητών στο μετρητή προγράμματος. Με τον τρόπο αυτόν φεύγουμε από το υποπρόγραμμα zero() και επιστρέφουμε στο πρόγραμμα ή υποπρόγραμμα που το κάλεσε.

Άσκηση 4:

Θεωρήστε τη μονάδα επεξεργασίας δεδομένων της απλοποιημένης αρχιτεκτονικής MIPS, στην οποία κάθε εντολή ολοκληρώνεται σε έναν κύκλο μηχανής. Τροποποιήστε την αρχιτεκτονική ώστε να υποστηρίζει έμμεση διευθυνσιοδότηση στη μνήμη μέσω καταχωρητή στις πράξεις της ΑΛΜ, όπως για παράδειγμα στην επόμενη εντολή:

add \$8, \$10, (\$7)

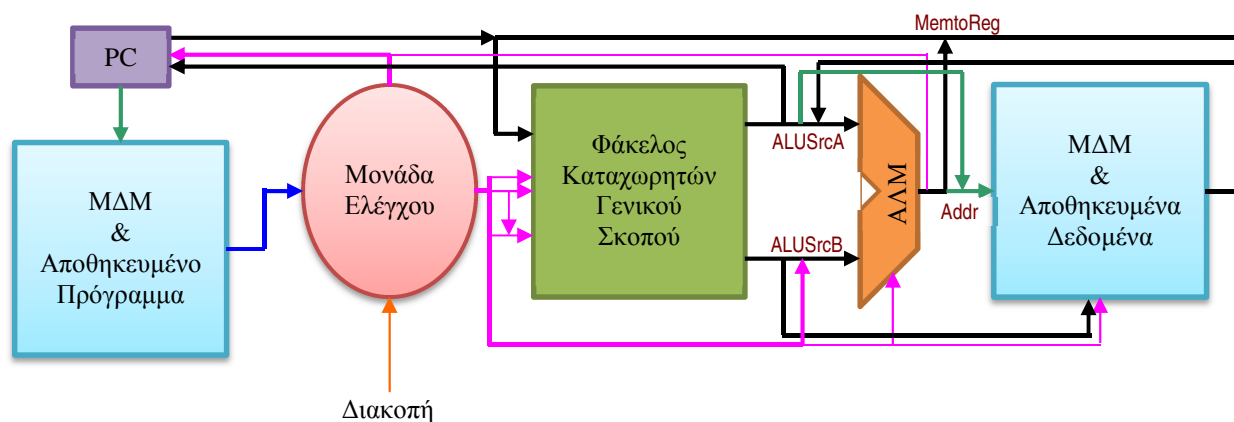
Το πρώτο τελούμενο και το αποτέλεσμα σ' αυτή τη διευθυνσιοδότηση είναι καταχωρητές γενικού σκοπού. Υποθέστε ότι η διευθυνσιοδότηση αυτή δεν υποστηρίζει μετατόπιση στην αναφορά, έτσι ώστε να μη χρειάζεται η ΑΛΜ για τον υπολογισμό της τελικής διεύθυνσης. Συμπληρώστε όποια στοιχεία ελέγχου θεωρείτε απαραίτητα για τη σωστή επιλογή εισόδων στην ΑΛΜ.

Απάντηση:

Σε μια εντολή όπως η “add \$8, \$10, (\$7)” θέλουμε ο φάκελος καταχωρητών γενικού σκοπού να στέλνει το αποτέλεσμα της μιας ανάγνωσης (το περιεχόμενο του \$10) στην ΑΛΜ και το αποτέλεσμα της άλλης ανάγνωσης (το περιεχόμενο του \$7) στη ΜΔΜ δεδομένων σε διεύθυνση προσπέλασης. Επίσης θέλουμε η ΜΔΜ δεδομένων να στέλνει το αποτέλεσμα της ανάγνωσης από τη μνήμη στην ΑΛΜ.

Έτσι, για να μπορέσουμε να έχουμε έμμεση διευθυνσιοδότηση στη μνήμη σε εντολές πράξεων της ΑΛΜ πρέπει: (α) η ΑΛΜ να δέχεται είσοδο από τη ΜΔΜ δεδομένων και (β) η ΜΔΜ δεδομένων να δέχεται είσοδο διεύθυνσης από το φάκελο καταχωρητών.

Λαμβάνοντας υπ' όψη τη νέα ροή πληροφορίας στη μονάδα επεξεργασίας δεδομένων, και υποθέτοντας ότι ο καταχωρητής έμμεσης διευθυνσιοδότησης κωδικοποιείται στο πεδίο rs της λέξης εντολής, θα πάρουμε το εξής διορθωμένο διάγραμμα:



Αυτό που πρέπει να βρεθεί είναι το πώς θα παρέμβουμε στον έλεγχο ροής για να υποστηρίξουμε τους νέους δρόμους ροής πληροφορίας. Έχοντας συμπεριλάβει στο διάγραμμα τους πολυπλέκτες ALUSrcB και MemtoReg που είχαμε εισάγει στην Άσκηση 2, τον πρώτο να επιλέγει τη δεύτερη είσοδο της ΑΛΜ από το φάκελο καταχωρητών ή από τη μονάδα ελέγχου (αγνοώντας τη δυνατότητα επιλογής επέκτασης μηδενικών αντί προσήμου για απλούστευση), και το δεύτερο να επιλέγει την είσοδο του φακέλου καταχωρητών από τη ΜΑΜ δεδομένων ή από την ΑΛΜ, θα χρειαστούμε δύο ακόμα πολυπλέκτες 2×1, ως εξής:

(α) Έναν πολυπλέκτη 2×1 (Addr) να επιλέγει την είσοδο διεύθυνσης της ΜΑΜ δεδομένων. Τιμή 1 για το σήμα επιλογής οδηγεί στη ροή της διεύθυνσης από το φάκελο καταχωρητών, ενώ τιμή 0 οδηγεί στη ροή που είχαμε και προηγουμένως, δηλαδή από την ΑΛΜ.

(β) Έναν πολυπλέκτη 2×1 (ALUSrcA) να επιλέγει την πρώτη είσοδο της ΑΛΜ. Τιμή 1 για το σήμα επιλογής οδηγεί στη ροή δεδομένων από τη ΜΑΜ δεδομένων, ενώ τιμή 0 οδηγεί στη ροή που είχαμε, δηλαδή από το φάκελο καταχωρητών.

Τα σήματα επιλογής για τους δύο νέους πολυπλέκτες παράγονται όπως και των υπολοίπων από τη μονάδα ελέγχου σαν αποτέλεσμα της φάσης αποκωδικοποίησης του κύκλου εντολής.

Για να επαληθεύσουμε ότι το παραπάνω σχήμα υποστηρίζει τη διευθυνσιοδότηση που θέλουμε, κι επίσης ότι δεν καταστρέφει τις υπάρχουσες διευθυνσιοδοτήσεις και ροές πληροφορίας, θα βρούμε με ποιες τιμές επιλογής για τους 4 πολυπλέκτες υλοποιείται κάθε μια μέθοδος διευθυνσιοδότησης και ροή πληροφορίας από όσες πιθανά επηρεάζονται από τις τροποποιήσεις που εισάγαμε στη μονάδα επεξεργασίας δεδομένων:

ALUSrcB	MemtoReg	Addr	ALUSrcA	Διευθυνσιοδότηση	Ροή πληροφορίας	Παράδειγμα
1	0	x	0	κατ' ευθείαν/άμεση	ΦΚ/ΜΕ→ΑΛΜ→ΦΚ	andi \$8,\$10,7
0	0	x	0	κατ' ευθείαν	ΦΚ/ΦΚ→ΑΛΜ→ΦΚ	sub \$8,\$10,\$7
0	0	1	1	κατ' ευθείαν/έμμεση	ΦΚ/(ΦΚ→ΜΔ)→ΑΛΜ→ΦΚ	or \$8,\$10,(\$7)
1	1	0	0	σχετική έμμεση	(ΦΚ/ΜΕ→ΑΛΜ)→ΜΔ→ΦΚ	lw \$8,10(\$7)
1	x	0	0	κατ' ευθείαν/σχετική έμμεση	ΦΚ/(ΦΚ/ΜΕ→ΑΛΜ)→ΜΔ	sw \$8,10(\$7)

όπου ΦΚ = Φάκελος Καταχωρητών, ΜΕ = Μονάδα Ελέγχου, ΜΔ = Μνήμη Δεδομένων.

Για παράδειγμα, η ήδη υπάρχουσα κατ' ευθείαν/άμεση διευθυνσιοδότηση, με ροή πληροφορίας από το ΦΚ και τη ΜΕ προς την ΑΛΜ και από την ΑΛΜ πίσω στο ΦΚ, επιλέγεται με τιμές επιλογής 1 για τον ALUSrcB (δεύτερη είσοδος της ΑΛΜ από τη ΜΕ), 0 για τον MemtoReg (είσοδος δεδομένων εγγραφής του ΦΚ από την ΑΛΜ), αδιάφορη για τον Addr

(αδιάφορη είσοδος διεύθυνσης της ΜΔ) και 0 για τον ALUSrcA (πρώτη είσοδος της ΑΛΜ από το ΦΚ).

Η νέα κατ' ευθείαν/έμμεση διευθυνσιοδότηση έχει ροή πληροφορίας από το ΦΚ στη ΜΔ για την έμμεση αναφορά, κι από εκεί στην ΑΛΜ, της οποίας η άλλη είσοδος λαμβάνεται από το ΦΚ, και η έξοδος στέλνεται στο ΦΚ. Οι τιμές επιλογής των πολυπλεκτών θα είναι 0 για τον ALUSrcB (δεύτερη είσοδος της ΑΛΜ από το ΦΚ), 0 για τον MemtoReg (είσοδος δεδομένων εγγραφής του ΦΚ από την ΑΛΜ), 1 για τον Addr (είσοδος διεύθυνσης της ΜΔ από το ΦΚ) και 1 για τον ALUSrcA (πρώτη είσοδος της ΑΛΜ από τη ΜΔ).

Τέλος η κατ' ευθείαν/σχετική έμμεση διευθυνσιοδότηση των εντολών αποθήκευσης στη μνήμη έχει ροή πληροφορίας από το ΦΚ και τη ΜΕ προς την ΑΛΜ για τον υπολογισμό της τελικής διεύθυνσης προσπέλασης, και από την ΑΛΜ και τη δεύτερη έξοδο του ΦΚ προς τη ΜΔ. Η ροή αυτή επιλέγεται από τους πολυπλέκτες με τιμές 1 για τον ALUSrcB (δεύτερη είσοδος της ΑΛΜ από τη ΜΕ), αδιάφορη για τον MemtoReg (δεν έχουμε εγγραφή στο ΦΚ), 0 για τον Addr (είσοδος διεύθυνσης της ΜΔ από την ΑΛΜ) και 0 για τον ALUSrcA (πρώτη είσοδος της ΑΛΜ από το ΦΚ). Η ροή από το ΦΚ προς την είσοδο δεδομένων της ΜΔ δε χρειάζεται επιλογή.

Η εισαγωγή των νέων πολυπλεκτών ανοίγει πρόσθετους διαδρόμους ροής πληροφορίας. Όμως μόνο οι παραπάνω ροές πληροφορίας είναι επιτρεπτές. Άλλες ροές είναι πιθανό να δημιουργούν κύκλωμα επανατροφοδότησης που δε μπορούμε να υποστηρίξουμε στην αρχιτεκτονική του ενός κύκλου μηχανής για κάθε κύκλο εντολής. Έτσι, δε μπορούμε να έχουμε τιμή επιλογής 0 για τον Addr και τιμή επιλογής 1 για τον ALUSrcA, αλλιώς θα έχουμε τον κύκλο $ΑΛΜ \rightarrow ΜΔ \rightarrow ΑΛΜ$ στη ροή πληροφορίας που θα έχει απρόβλεπτο αποτέλεσμα. Τον κύκλο αυτόν θα παίρναμε για παράδειγμα, αν θέλαμε μαζί με την έμμεση διευθυνσιοδότηση σε εντολές της ΑΛΜ να υποστηρίζαμε και μετατόπιση στην τελική διεύθυνση. Εκτός από τη λύση της χρησιμοποίησης δεύτερης ΑΛΜ που να εκτελεί την απαιτούμενη πρόσθεση για τον υπολογισμό της τελικής διεύθυνσης προσπέλασης, θα έπρεπε να επιτρέψουμε στη μοναδική ΑΛΜ να χρησιμοποιείται δύο φορές σε κάθε κύκλο εντολής. Τότε όμως η επαναχρησιμοποίηση της ΑΛΜ στον ίδιο κύκλο μηχανής δημιουργεί το μη ελεγχόμενο κύκλωμα επανατροφοδότησης. Η μοναδική λύση για έλεγχο κυκλικής ροής πληροφορίας είναι η υλοποίηση του κύκλου εντολής με πολλαπλούς κύκλους μηχανής, ώστε κάθε φάση του κύκλου εντολής να εκτελείται σε διαφορετικό κύκλο μηχανής.