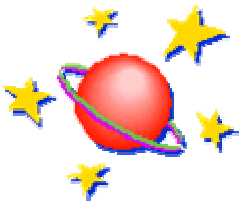

Ενότητα 6η : *Η Αρχιτεκτονική του υπολογιστή MIPS*

Σκοπός **Ο σκοπός της ενότητας αυτής είναι να παρουσιάσει την αρχιτεκτονική του υπολογιστή MIPS.**

Προσδοκώμενα Αποτελέσματα **Όταν θα έχετε μελετήσει την ενότητα, θα είστε σε θέση να:**

αναφέρετε τους καταχωρητές και τους τύπους δεδομένων του MIPS,

παρουσιάζετε τους τρόπους διευθυνσιοδότησης που χρησιμοποιούνται στον MIPS,

αναφέρετε τους τρόπους κωδικοποίησης των εντολών του MIPS,

απαριθμείτε τις εντολές που υποστηρίζει ο MIPS,

εξηγείτε την εκτέλεση των εντολών που υποστηρίζει ο MIPS,

χρησιμοποιείτε τον προσομοιωτή PCSPim για την εκτέλεση προγραμμάτων σε γλώσσα assembly.

ευθυγραμμισμένα δεδομένα

Καταχωρητές και τύποι δεδομένων στον υπολογιστή MIPS

Ο MIPS διαθέτει 32 καταχωρητές γενικής χρήσης των 32 δυαδικών ψηφίων (32 bits) οι οποίοι ονομάζονται $0, $1, … , $31 αντίστοιχα και χρησιμοποιούνται ως εξής:

- Ο καταχωρητής $0 έχει πάντα την τιμή 0.
- Οι καταχωρητές $1, $26 και $27 χρησιμοποιούνται από το λειτουργικό σύστημα και δεν χρησιμοποιούνται από τους μεταγλωττιστές (compilers) ή από τα προγράμματα.
- Οι καταχωρητές $2 και $3 χρησιμοποιούνται για να επιστρέφουν τιμές από συναρτήσεις.
- Οι καταχωρητές $4-$7 χρησιμοποιούνται για την αποθήκευση πληροφοριών.
- Οι καταχωρητές $8-$15 και οι $24, $25 χρησιμοποιούνται για την προσωρινή αποθήκευση δεδομένων.
- Οι καταχωρητές $16-$23 χρησιμοποιούνται για την αποθήκευση τιμών οι οποίες μπορεί να χρησιμοποιηθούν κατά τη διάρκεια μιας κλήσης διαδικασιών.
- Ο καταχωρητής $28 είναι ένας καταχωρητής δείκτης, ο οποίος δείχνει στη μέση μιας περιοχής δεδομένων της στατικής μνήμης δεδομένων των 64K.
- Ο καταχωρητής $29 είναι ο δείκτης στοίβας.
- Ο καταχωρητής $30 είναι ο δείκτης πλαισίου (frame pointer), και ο $31 χρησιμοποιείται μόνο στην περίπτωση της εντολής jal για την επιστροφή της διεύθυνσης μετά από κλήση διαδικασίας.

Ο υπολογιστής MIPS περιέχει 32 επιπλέον ειδικούς καταχωρητές για τα δεδομένα κινητής υποδιαστολής, οι οποίοι ονομάζονται καταχωρητές κινητής υποδιαστολής και συμβολίζονται με: $f0, $f1, …, $f31. Μόνο οι 16 από τους καταχωρητές κινητής υποδιαστολής του MIPS χρησιμοποιούνται για τα δεδομένα μονής ακρίβειας και είναι οι : $f0, $f2, …, $f30. Για τα δεδομένα κινητής υποδιαστολής διπλής ακρίβειας, οι 16 αυτοί καταχωρητές σχηματίζουν 8 ζευγάρια καταχωρητών, που χρησιμοποιούνται για τους υπολογισμούς. Οι καταχωρητές κινητής υποδιαστολής με μονό αριθμό ($f1, $f3, …, $f31), χρησιμοποιούνται για την φόρτωση ή αποθήκευση των λιγότερο σημαντικών ψηφίων στους αριθμούς κινητής υποδιαστολής των 64 bits.

Οι τύποι δεδομένων του υπολογιστή MIPS είναι:

- 1 byte (8 bits), 2 bytes (16 bits – μισή λέξη) και 4 bytes (32 bits - λέξη) για ακέραια δεδομένα.
- 4 bytes (32 bits) απλής ακρίβειας και 8 bytes (64 bits) διπλής ακρίβειας για κλασματικούς αριθμούς (κινητής υποδιαστολής).

Όλες οι εντολές στον MIPS έχουν μέγεθος 4 bytes.

Τρόποι διευθυνσιοδότησης του υπολογιστή MIPS

Στον υπολογιστή MIPS υπάρχουν οι παρακάτω τρόποι διευθυνσιοδότησης:

Τρόπος διευθυνσιοδότησης	Υπολογισμός διεύθυνσης
Απευθείας	Ο τελεστέος είναι μία σταθερά με γνωστή διεύθυνση.
Καταχωρητή	Ο τελεστέος είναι το περιεχόμενο ενός καταχωρητή.
Έμμεσος καταχωρητή με μετατόπιση	Ο τελεστέος βρίσκεται στη διεύθυνση μνήμης, η οποία προκύπτει από το άθροισμα του περιεχομένου του καταχωρητή και μίας σταθεράς.

Στην πραγματικότητα ο υπολογιστής MIPS χρησιμοποιεί μόνο ένα τρόπο διευθυνσιοδότησης μνήμης, τον c(rx), όπου η διεύθυνση μνήμης υπονοείται και χρησιμοποιείται το άθροισμα του απ' ευθείας δεδομένου c και τον καταχωρητή rx σαν διεύθυνση.

Παράδειγμα:

Έστω ότι θέλουμε να προσθέσουμε τη σταθερά 4 στον καταχωρητή $29, η οποία βρίσκεται στη διεύθυνση AddrConstant4. Για να γίνει αυτό στον υπολογιστή MIPS θα πρέπει να εκτελεστεί η παρακάτω ακολουθία εντολών:

```
lw    $24, AddrConstant4($0)
add   $29, $29, $24
```

Με την πρώτη εντολή η οποία είναι μια εντολή φόρτωσης, αρχικά φορτώνεται η σταθερά 4 στον καταχωρητή $24. Στη συνέχεια, με την εντολή add προστίθεται το περιεχόμενο του καταχωρητή $24 (η σταθερά 4) με το περιεχόμενο του καταχωρητή $29 και το αποτέλεσμα αποθηκεύεται στον καταχωρητή $29 Παρατηρούμε ότι χρειάζονται δύο εντολές για να εκτελεστεί αυτή η πράξη.
Προκειμένου να αποφύγουμε τις εντολές επικοινωνίας με τη μνήμη (φόρτωση ή αποθήκευση), μπορεί να χρησιμοποιηθεί ο απευθείας τρόπος διευθυνσιοδότησης. Επομένως, για την εντολή add του παραδείγματος χρησιμοποιείται ένας τελεστέος ο οποίος είναι σταθερά και ονομάζεται addi (add immediate). Για να προσθέσουμε λοιπόν τη σταθερά 4 στον καταχωρητή $29 έχουμε:

```
addi  $29, $29, 4
```

Επομένως με τη χρήση του απευθείας τρόπου διευθυνσιοδότησης, το περιεχόμενο του καταχωρητή $29 προστίθεται με τη σταθερά 4 και το αποτέλεσμα αποθηκεύεται στον καταχωρητή $29, χωρίς να μεσολαβήσει επικοινωνία με τη μνήμη.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

- **Να προσδιορίσετε ποιος τρόπος διευθυνσιοδότησης χρησιμοποιείται στην παρακάτω εντολή πρόσθεσης του υπολογιστή MIPS: add $1, $2, $3**
- **Ποιες είναι οι αντίστοιχες εντολές που εκτελούν την ίδια λειτουργία, στους υπολογιστές DLX και VAX; (Θεωρήστε ότι οι πηγαίοι τελεστέοι έχουν μέγεθος 4 bytes και η μετατόπιση είναι μεγέθους 2 bytes.)**

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

- **Στην εντολή αυτή χρησιμοποιείται ο τρόπος διευθυνσιοδότησης καταχωρητή, αφού ο ένας πηγαίος τελεστέος είναι καταχωρητής.**
- **DLX: ADD R1, R2, R3**
  VAX: ADDL3 W R2, R3, R1

Παράδειγμα:

Έστω η παρακάτω εντολή φόρτωσης δεδομένων από τη μνήμη: lw $1, 100($2)
Η εντολή αυτή χρησιμοποιεί τον τρόπο διευθυνσιοδότησης: έμμεσος καταχωρητή με αυτοαύξηση. Η εκτέλεση της εντολής αυτής μεταφέρει δεδομένα από μία θέση μνήμης σε έναν καταχωρητή. Η διεύθυνση της μνήμης που υπάρχουν τα δεδομένα υπολογίζεται από το άθροισμα της τιμής 100 και το πριεχόμενο του καταχωρητή $2. Το αποτέλεσμα αποθηκεύεται στον καταχωρητή $1.

Κωδικοποίηση των εντολών του MIPS

Ο τρόπος με τον οποίο κωδικοποιούνται οι εντολές στον υπολογιστή MIPS απεικονίζεται στο σχήμα 2.6.1. Όπως φαίνεται και στο σχήμα, η εντολή αποτελείται από 32 bits, δηλαδή όσο είναι το μέγεθος μιας λέξης δεδομένων.

Τα πεδία της εντολής είναι διαφορετικά ανάλογα με το είδος της εντολής που κωδικοποιείται και τον τρόπο διευθυνσιοδότησης που χρησιμοποιείται.

Για παράδειγμα, οι αριθμητικές και λογικές εντολές με διευθυνσιοδότηση καταχωρητή χωρίζονται σε έξι πεδία τα οποία είναι:

- op: κωδικός λειτουργίας της εντολής,
- rs: ο πρώτος πηγαίος καταχωρητής,
- rt: ο δεύτερος πηγαίος καταχωρητής,
- rd: ο καταχωρητής προορισμού,
- shamt: ψηφία ολίσθησης,
- funct: λειτουργία, το πεδίο αυτό καθορίζει τη λειτουργία που θα εκτελεστεί από την εντολή σύμφωνα με το πεδίο op.

6	5	5	5	5	6
op	rs	rt	rd	shamt	funct

Σχήμα 1 – Τα πεδία της εντολής υπολογιστή MIPS.

Οι εντολές του MIPS κωδικοποιούνται σε έναν από τους παρακάτω τύπους:

- **Αριθμητικές και λογικές εντολές (τύπου R)**
- **Εντολές επικοινωνίας με τη μνήμη**
- **Διακλαδώσεις**

Αριθμητικές και λογικές εντολές

Στις αριθμητικές και λογικές εντολές περιλαμβάνονται προσθέσεις, αφαιρέσεις, πολλαπλασιασμοί, διαιρέσεις και λογικές πράξεις (and, or, κλπ.) για όλους τους τύπους δεδομένων του υπολογιστή MIPS.

Παράδειγμα:

Πώς κωδικοποιούνται και εκτελούνται οι παρακάτω εντολές του MIPS;

and rd, rs, rt
addi rt, rs, Imm

Απάντηση:

Στην περίπτωση της πρώτης εντολής ο τρόπος διευθυνσιοδότησης είναι καταχωρητή και η λειτουργία που πρόκειται να εκτελεστεί είναι το λογικό ΚΑΙ. Δηλαδή, θα εκτελεστεί η πράξη στους πηγαίους καταχωρητές rs και rt και το αποτέλεσμα θα αποθηκευτεί στον καταχωρητή προορισμού rd. Επομένως η εντολή κωδικοποιείται ως εξής:

and rd, rs, rt

op	rs	rt	rd	0	Function (and)
6	5	5	5	5	6

bits

Η δεύτερη εντολή χρησιμοποιεί τον απευθείας τρόπο διευθυνσιοδότησης με αποτέλεσμα τα δύο τελευταία πεδία της εντολής να αντικαθίστανται με τη σταθερά Immediate.

addi rt, rs, Imm

op	rs	rt	Imm
6	5	5	16

bits

Εντολές επικοινωνίας με τη μνήμη

Στις εντολές αυτές περιλαμβάνονται οι εντολές φόρτωσης δεδομένων από τη μνήμη και οι εντολές αποθήκευσης δεδομένων σε καταχωρητές.

Παράδειγμα:

Πως κωδικοποιούνται και εκτελούνται οι παρακάτω εντολές του MIPS;

lw $1, 100($0)
sw $1, 100($2)

Απάντηση:

lw $1, 100($0): Η εντολή αυτή χρησιμοποιεί τον τρόπο κωδικοποίησης: έμμεσος καταχωρητή με μετατόπιση. Η μετατόπιση είναι η σταθερά 100 και βρίσκεται στο τελευταίο πεδίο της εντολής. Με την εντολή αυτή, η διεύθυνση από την οποία θα φορτωθούν τα δεδομένα στον καταχωρητή $1υπολογίζεται ως εξής: προστίθεται το περιεχόμενο του καταχωρητή $0 (είναι πάντα 0) με την τιμή 100.

lw $1, 100($0)

op	$0	$1	100
6	5	5	16 bits

sw $1, 100($2) : Η εντολή αυτή χρησιμοποιεί τον τρόπο κωδικοποίησης: έμμεσος καταχωρητή με μετατόπιση. Η μετατόπιση είναι η σταθερά 100 και βρίσκεται στο τελευταίο πεδίο της εντολής, όπως και στην εντολή φόρτωσης. Με την εντολή αυτή το περιεχόμενο του καταχωρητή $1 αποθηκεύεται σε μία θέση μνήμης. Η διεύθυνση της μνήμης υπολογίζεται με τον ίδιο τρόπο όπως και στην προηγούμενη εντολή του παραδείγματος αυτού.

lw $1, 100($2)

op	$2	$1	100
6	5	5	16 bits

Τα δεδομένα ονομάζονται ευθυγραμμισμένα εάν η διεύθυνση της μνήμης που περιέχει τα δεδομένα, είναι πολλαπλάσια του μεγέθους των δεδομένων σε bytes. Οι περισσότερες εντολές φόρτωσης/αποθήκευσης εκτελούνται σε **ευθυγραμμισμένα δεδομένα**. Οι εντολές του MIPS για το χειρισμό των ευθυγραμμισμένων δεδομένων είναι: lwl, lws, swl και swl.

Διακλαδώσεις

Οι εντολές διακλάδωσης περιλαμβάνουν τις εντολές διακλάδωσης με συνθήκη και τις εντολές μεταπήδησης.

Παράδειγμα:

Πως κωδικοποιούνται και εκτελούνται οι παρακάτω εντολές του MIPS;

beq rs, rt, label

j label

(Το label είναι μία σταθερά)

Απάντηση:

beq rs, rt, label : Με την εντολή αυτή γίνεται σύγκριση των περιεχομένων των καταχωρητών rs και rt. Στην περίπτωση που είναι ίσα τότε η διεύθυνση της επόμενης εντολής υπολογίζεται ως εξής: αυξάνεται το περιεχόμενο του PC κατά 4 + τη σταθερά label. Επομένως η επόμενη προς εκτέλεση εντολή βρίσκεται στην διεύθυνση που υπολογίστηκε. Στην περίπτωση που το περιεχόμενο των δύο καταχωρητών δεν είναι ίσο, τότε η επόμενη εντολή βρίσκεται στη διεύθυνση που υπολογίζεται με την αύξηση του PC κατά 4. Η εντολή κωδικοποιείται ως εξής:

beq rs, rt, label

op	rs	rt	label
6	5	5	16

bits

j label : Η εντολή αυτή είναι μια εντολή μεταπήδησης, η οποία αλλάζει τη ροή εκτέλεσης του προγράμματος, μεταφέροντάς το στη διεύθυνση label. Η εντολή αυτή κωδικοποιείται ως εξής: τα πρώτα 6 bits αποτελούν τον κωδικό λειτουργίας της εντολής και τα υπόλοιπα 26 αποτελούν το πεδίο της διεύθυνσης.

j label

op	label
6	26

bits

Σύνολο εντολών του MIPS

Παρακάτω, παρουσιάζεται ο συγκεντρωτικός πίνακας που περιλαμβάνει όλους του τύπους εντολών του MIPS καθώς και τη σημασία της κάθε εντολής.

Τύπος εντολής/ κωδικός πράξης	**Σημασία εντολής**
Εντολές επικοινωνίας με τη μνήμη	**Μετακίνηση δεδομένων μεταξύ καταχωρητών και μνήμης.**
lb, lbu, sb	Φόρτωση byte, φόρτωση byte χωρίς πρόσημο, αποθήκευση byte.
lh, lhu ,sh	Φόρτωση μισής λέξης, φόρτωση μισής λέξης χωρίς πρόσημο, αποθήκευση μισής λέξης.
lw, sw	Φόρτωση λέξης, αποθήκευση λέξης (σε/από καταχωρητή ακεραίων).
lwc1, ldc1, swc1, swc1	Φόρτωση καταχωρητή κινητής υποδιαστολής απλής ακρίβειας, φόρτωση καταχωρητή κινητής υποδιαστολής διπλής ακρίβειας, αποθήκευση καταχωρητή κινητής υποδιαστολής απλής ακρίβειας, αποθήκευση καταχωρητή κινητής υποδιαστολής διπλής ακρίβειας.
Αριθμητικές/ λογικές	**Πράξεις σε ακέραια ή λογικά δεδομένα που βρίσκονται σε καταχωρητές γενικής χρήσης.**
add, addi, addu, addui	Πρόσθεση, απευθείας πρόσθεση (όλα τα απευθείας δεδομένα είναι 16 bits) με πρόσημο και χωρίς.
sub, subi, subu, subui	Αφαίρεση, απευθείας αφαίρεση προσημασμένη και μη.
mult, multu, div, divu	Πολλαπλασιασμός και διαίρεση, προσημασμένα και μη. Οι τελεστέοι πρέπει να βρίσκονται σε καταχωρητές κινητής υποδιαστολής. Όλες οι πράξεις παίρνουν και δίνουν τιμές 32-bit.
and, andi	And, απευθείας and.
or, ori, xor, xori	Or, απευθείας or, αποκλειστικό or, απευθείας αποκλειστικό or.
lui	Φόρτωση στα δύο πιο σημαντικά bytes ενός καταχωρητή μια απευθείας τιμή των 16 bits.
sslv, srlv, srav, sll, srl, sra	Ολισθήσεις: απευθείας (S_I) και μεταβλητή (S_) μορφή. Οι ολισθήσεις είναι αριστερές λογικές, δεξιές λογικές, δεξιές αριθμητικές.

slt	Η συνθήκη: "_" μπορεί να είναι LT,GT, LE, GE, EQ, NE
Διακλαδώσεις	**Υπό συνθήκη διακλαδώσεις και μεταπηδήσεις. PC-σχετικές ή μέσω καταχωρητών**
beq, bne	Διακλάδωση αν το περιεχόμενο του καταχωρητή είναι ίσο/άνισο με το περιεχόμενο κάποιου άλλου καταχωρητή.
j, jr	Μεταπηδήσεις: 26-bit offset από PC+4 (J) ή στόχος σε καταχωρητή (jr).
jal, jalr	Μεταπήδηση και σύνδεση: σώσε PC+4 στον $31,στόχος είναι PC-σχετικός (jal) ή ένας καταχωρητής (jal).
break	Μεταφορά στο λειτουργικό σύστημα σε ένα διάνυσμα διεύθυνσης
rfe	Επιστροφή στον χρησιμοποιούμενο κώδικα από μία εξαίρεση. Επαναφύλαξη του Τρόπου του χρήστη.

Ο προσομοιωτής PCSPim

Ο προσομοιωτής Spim προσομοιώνει την εκτέλεση των εντολών του υπολογιστή MIPS, εκτελώντας προγράμματα στη γλώσσα προγραμματισμού assembly. Ο προσομοιωτής Spim βρίσκεται στην ιστοσελίδα:

http://hermes.di.uoa.gr/PCSpim.zip

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

Για περισσότερες λεπτομέρειες για τη γλώσσα assembly και τον τρόπο λειτουργίας του προσομοιωτή PCSPim κρίνετε σκόπιμο να συμβουλευτείτε το βιβλίο: "Computer Organization: The Hardware/Software Interface", Patterson & Hennessy, 2nd Ed., Morgan Kaufmann Publishers.

Παράδειγμα

Το παρακάτω πρόγραμμα σε κώδικα assembly εκτελεί την πρόσθεση δύο αριθμών. Συγκεκριμένα, προσθέτει το περιεχόμενο των διευθύνσεων vb, vc και αποθηκεύει το αποτέλεσμα στο διεύθυνση va).

```
        .data 0x10000000        (ορίζει τη διεύθυνση από την οποία θα
                                 αρχίσουν να αποθηκεύονται τα δεδομένα)

va:     .word 0x12345678        (αποθηκεύει στη διεύθυνση που ορίζεται
                                 από την εντολή .data, δηλαδή στην
                                 0x10000000, την τιμή 0x12345678 και
                                 δίνει στη διεύθυνση αυτή την ετικέτα
                                 va)
vb:     .word 0x01020304        (αποθηκεύει στη διεύθυνση 0x10000004
                                 την τιμή 0x01020304 και δίνει στη
                                 διεύθυνση αυτή την ετικέτα vb)
vc:     .word 0x04030201        (αποθηκεύει στη διεύθυνση 0x10000008
                                 την τιμή 0x04030201 και δίνει στη
                                 διεύθυνση αυτή την ετικέτα vc)

      .text                     (ορίζεται η αρχή των εντολών)
      .globl main               (χρησιμοποιείται πάντα στην αρχή των
                                 προγραμμάτων του προσομοιωτή Spim
                                 και δηλώνει πως το συγκεκριμένο
                                 πρόγραμμα μπορεί να χρησιμοποιηθεί από
                                 διαφορετικά αρχεία του προσομοιωτή)
```

```
main:                          (ξεκινάει το κυρίως πρόγραμμα)
    la $t0, vb                 (φορτώνεται στον καταχωρητή $t0
                                η διεύθυνση vb(όχι το περιεχόμενό
                                της, αλλά η ίδια η διεύθυνση))
    lw $t0, 0x0($t0)           (φορτώνονται στον καταχωρητή $t0
                                τα δεδομένα που βρίσκονται στη
                                διεύθυνση μνήμης που περιέχονται
                                στον καταχωρητή $t0)
    la $t1, vc                 (φορτώνεται στον καταχωρητή $t1
                                η διεύθυνση vc)
    lw $t1, 0x0($t1)           (φορτώνονται στον καταχωρητή $t1
                                τα δεδομένα που βρίσκονται στη
                                διεύθυνση μνήμης που περιέχεται
                                στον καταχωρητή $t1)
    add $t1, $t0, $t1          (γίνεται η πρόσθεση των περιεχομένων
                                των καταχωρητών $t0 και $t1 και το
                                αποτέλεσμα αποθηκεύεται στον
                                καταχωρητή $t1)
    la $t0, va                 (φορτώνεται στον καταχωρητή $t0
                                η διεύθυνση va)

    sw $t1, 0x0($t0)           (αποθήκευση του περιεχομένου του
                                καταχωρητή $t1 στη διεύθυνση μνήμης
                                που περιέχεται στον καταχωρητή $t0)
    li $v0, 10                (οι δύο αυτές εντολές τερματίζουν το
    syscall                    πρόγραμμα στον προσομοιωτή Spim)
```

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

α)Να εξηγήσετε την εκτέλεση των παρακάτω εντολών του υπολογιστή MIPS:

- **lw $t0, 0x0($t0)**
- **add $t1, $t0, $t1**
- **sw $t1, 0x0($t0)**

Να συγκρίνετε την απάντησή σας με τις απαντήσεις που δίνονται στο προηγούμενο παράδειγμα.

β)Να εκτελέσετε το πρόγραμμα του παραπάνω παραδείγματος στον προσομοιωτή Spim, (με τα δεδομένα που έχει), και να σχολιάσετε τα αποτελέσματα.

Ο προσομοιωτής Spim βρίσκεται στη σελίδα:

http://hermes.di.uoa.gr/PCSpim.zip

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Δίνεται το παρακάτω πρόγραμμα σε κώδικα assembly.

```
          .data    0x10000000
a:        .word    0x1, 0x2, 0x4, 10
n:        .word    0x4
sum:      .space   4

          .text
          .globl main

main:    la $t0, sum
         sw $0, 0($t0)
         la  $t1, n
         lw $t1, 0($t1)
         add $t2, $0, $0
         la $t3, a

brox:    bge    $t2, $t1, telos

         add    $t4, $t2, $t2
         add    $t4, $t4, $t4
         add    $t4, $t3, $t4
         lw     $t4, 0x0($t4)

         lw     $t5, 0x0($t0)
         add    $t5, $t5, $t4
         sw     $t5, 0x0($t0)

         addi $t2, $t2, 1
         beq  $0, $0, brox

telos:   li    $v0, 10
         syscall
```

α)Να εξηγήσετε την εκτέλεση των εντολών του προγράμματος.
β)Να περιγράψετε τη λειτουργία του προγράμματος.
γ)Να εκτελέσετε το παραπάνω πρόγραμμα στον προσομοιωτή Spim και να σχολιάσετε τα αποτελέσματα. Ο προσομοιωτής Spim βρίσκεται στη σελίδα:

http://hermes.di.uoa.gr/PCSpim.zip

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4

α)

```
        .data    0x10000000          (ορίζει τη διεύθυνση από
                                      την οποία θα αρχίσουν να
                                      αποθηκεύονται τα δεδομένα)
a:      .word    0x1, 0x2, 0x4, 10   (αποθηκεύει στις διευθύνσεις
                                      0x10000000, 0x10000004,
                                      0x10000008, 0x1000000C
                                      τις τιμές των αριθμών
                                      (0x1,0x2,0x4,10) αντίστοιχα,
                                      και ονομάζει τη διεύθυνση
                                      0x10000000 a)
n:      .word    0x4                 (αποθηκεύει στη διεύθυνση
                                      0x100000010 την τιμή 0x4 και
                                      ονομάζει αυτή την διεύθυνση n)
sum:    .space   4                   (παραχωρεί χώρο 4 bytes στη
                                      διεύθυνση 0x100000014 και
                                      ονομάζει αυτή την διεύθυνση
                                      sum)

        .text                  (ορίζεται η αρχή των εντολών)
        .globl main            (χρησιμοποιείται πάντα στην αρχή των
                               προγραμμάτων του προσομοιωτή Spim
                               και δηλώνει πως το συγκεκριμένο
                               πρόγραμμα μπορεί να χρησιμοποιηθεί από
                               διαφορετικά αρχεία του προσομοιωτή)

main:                          (ξεκινάει το κυρίως πρόγραμμα)
       la $t0, sum             (φορτώνεται στον καταχωρητή $t0 η
                                διεύθυνση sum)
       sw $0, 0($t0)           (αποθήκευση της τιμής 0 στη διεύθυνση
                                μνήμης που περιέχεται στον καταχωρητή
                                $t0)
       la  $t1, n              (φορτώνεται στον καταχωρητή $t1 η
                                διεύθυνση n)
       lw $t1, 0($t1)          (φορτώνονται στον καταχωρητή $t1, τα
                                δεδομένα που βρίσκονται στη διεύθυνση
                                μνήμης που περιέχεται στον καταχωρητή
                                $t1)
       add $t2, $0, $0         (ο καταχωρητής $t2 παίρνει την
                                τιμή 0)
       la $t3, a               (φορτώνεται στον καταχωρητή $t3
                                η διεύθυνση a)
```

```
brox:                                   (αρχίζει ο βρόγχος)
        bge    $t2, $t1, telos          (ελέγχεται αν το περιεχόμενο του
                                         καταχωρητή $t2 είναι μεγαλύτερο ή
                                         ίσο με το περιεχόμενο του $t2 και
                                         αν ισχύει βγαίνει από το βρόγχο)
        add    $t4, $t2, $t2            (γίνεται η πρόσθεση των περιεχομένων
                                         των καταχωρητών $t2 και $t2 και το
                                         αποτέλεσμα αποθηκεύεται στον
                                         καταχωρητή $t4)
        add    $t4, $t4, $t4            (γίνεται η πρόσθεση των περιεχομένων
                                         των καταχωρητών $t4 και $t4 και το
                                         αποτέλεσμα αποθηκεύεται στον
                                         καταχωρητή $t4)
        add    $t4, $t3, $t4            (γίνεται η πρόσθεση των περιεχομένων
                                         των καταχωρητών $t3 και $t4 και το
                                         αποτέλεσμα αποθηκεύεται στον
                                         καταχωρητή $t4)
        lw     $t4, 0x0($t4)            (φορτώνονται στον καταχωρητή $t4 τα
                                         δεδομένα που βρίσκονται στη
                                         διεύθυνση μνήμης που περιέχεται
                                         στον καταχωρητή $t4)
        lw     $t5, 0x0($t0)            (φορτώνονται στον καταχωρητή $t5 τα
                                         δεδομένα που βρίσκονται στη
                                         διεύθυνση μνήμης που περιέχεται
                                         στον καταχωρητή $t0)
        add    $t5, $t5, $t4            (γίνεται η πρόσθεση των περιεχομένων
                                         των καταχωρητών $t4 και $t5 και το
                                         αποτέλεσμα αποθηκεύεται στον
                                         καταχωρητή $t5)
        sw     $t5, 0x0($t0)            (αποθήκευση του περιεχομένου του
                                         καταχωρητή $t5 στη διεύθυνση μνήμης
                                         που περιέχεται στον καταχωρητή
                                         $t0)
        addi $t2, $t2, 1                (γίνεται η πρόσθεση του περιεχομένου
                                         του καταχωρητή $t2 με το 1 και το
                                         αποτέλεσμα αποθηκεύεται στον
                                         καταχωρητή $t2)
        beq  $0, $0, brox               (μεταφέρει τη ροή του προγράμματος
                                         στο brox)
telos:  li    $v0, 10                   (οι δύο αυτές εντολές τερματίζουν
        syscall                          το πρόγραμμα στον προσομοιωτή SPim)
```

Ανακεφαλαιώνοντας λοιπόν …

Ο υπολογιστής MIPS υποστηρίζει όλους του συνήθεις τύπους δεδομένων, ενώ διαθέτει 32 καταχωρητές γενικής χρήσης και 32 επιπλέον ειδικούς καταχωρητές για τα δεδομένα κινητής υποδιαστολής.

Οι τρόποι διευθυνσιοδότησης στον MIPS είναι: απευθείας, καταχωρητή και έμμεσος καταχωρητή με μετατόπιση.

Τα πεδία των εντολών είναι διαφορετικά ανάλογα με το είδος της εντολής που κωδικοποιείται και τον τρόπο διευθυνσιοδότησης που χρησιμοποιείται.

Το σύνολο εντολών που υποστηρίζει ο υπολογιστής MIPS αποτελείται από: τις αριθμητικές και λογικές εντολές, τις εντολές επικοινωνίας με τη μνήμη και τις εντολές διακλάδωσης.

Για την εκτέλεση προγραμμάτων του υπολογιστή MIPS σε γλώσσα assembly, χρησιμοποιείται ο προσομοιωτής PCSPim.