Arithmetic Operations part B

Η γλώσσα MIPS (Microprocessor without Interlocked Pipeline Stages) assembly περιλαμβάνει λογικές οδηγίες (logical instructions) για την εκτέλεση λειτουργιών bitwise.

B. Logical Instructions:

1. and

and $destination, $source 1, $source 2

π.χ. : and $t0, $t1, $t2 # $t0 = $t1 & $t2

Εξήγηση: Εάν $t1 = 110010 και $t2 = 101100,

τότε το αποτέλεσμα σε $t0 θα είναι: $t0 = 100100

input	output	result
0	0	0
0	1	0
1	0	0
1	1	1

2. or

or $destination, $source 1, $source 2

π.χ. : or $t0, $t1, $t2 # $t0 = $t1 ή $t2

Εάν $t1 = 110010 και $t2 = 101100,

τότε το αποτέλεσμα σε $t0 θα είναι:

$t0 = 111110

input	output	result
0	0	0
1	0	1
0	1	1
1	1	1

3. andi

andi $destination, $source, άμεση

Εκτελεί μια λειτουργία and μεταξύ ενός καταχωρητή και μιας άμεσης τιμής.
π.χ.: andi $t0, $t1, 0xF # $t0 = $t1 & 0xF

4. ori

ori $destination, $source, άμεση

Εκτελεί μια λειτουργία or μεταξύ ενός καταχωρητή και μιας άμεσης τιμής. $t0, $t1, π.χ.: 0xA # $t0 = $t1 | 0xA

(5. xor , 6. nor)

5. Shift Left Logical (sll)

Μετατοπίζει τα bit ενός καταχωρητή αριστερά από έναν καθορισμένο αριθμό θέσεων.

sll $destination, $source, shift_amount

$destination = $source shifted left by shift_amount bits

Μετατοπίζεται το περιεχόμενο του καταχωρητή source αριστερά από την απόσταση που υποδεικνύεται από το shift_amount και καταλήγει στον καταχωρητή destination.

Παράδειγμα: sll $t0, $t1, 2 # μετατόπιση $t1 αριστερά κατά 2 bit και αποθηκεύεται στον $t0
Εάν $t1 = 00000001, τότε η μετατόπιση προς τα αριστερά κατά 2 bit θα έχει ως αποτέλεσμα:
$t0 = 00000100

6. **Shift Right Logical (srl)**

Μετατοπίζει τα bit ενός καταχωρητή δεξιά από έναν καθορισμένο αριθμό θέσεων.

srl $destination, $source, shift_amount

$destination = $source shifted right by shift_amount bits

Μετατοπίζεται το περιεχόμενο του καταχωρητή source δεξιά από την απόσταση που υποδεικνύεται από το shift_amount και καταλήγει στον καταχωρητή destination.

Παράδειγμα: srl $t0, $t1, 2 # μετατόπιση $t1 δεξιά κατά 2 bit και αποθηκεύεται στον $t0

Εάν $t1 = 00000100, τότε η μετατόπιση προς τα δεξιά κατά 2 bit θα έχει ως αποτέλεσμα: $t0 = 00000001

Γ. Comparison Instructions:

1. **Set on Less Than (slt)**

slt $destination, $source1, $source2

Ορίζει τον καταχωρητή destination σε 1 εάν ο καταχωρητής source1 είναι μικρότερο από το source2, διαφορετικά ορίζει 0. Δηλαδή $destination = 1 αν $source1 < $source2, διαφορετικά 0

Παράδειγμα: slt $t0, $t1, $t2 # $t0 = 1 αν $t1 < $t2, αλλιώς 0

Αν $t1 = 3 και $t2 = 5, τότε: $t0 = 1 (γιατί 3 < 5)

2. Set Less Than Immediate (slti)

slti $destination, $source1, imm

Ορίζει τον καταχωρητή destination σε 1 εάν ο καταχωρητής source1 είναι μικρότερο από τον αριυμό immediate, διαφορετικά ορίζει 0. Δηλαδή $destination = 1 αν $source1 <immediate, διαφορετικά 0

Παράδειγμα: slt $t0, $t1, 5 # $t0 = 1 αν $t1 < 5, αλλιώς 0

Αν $t1 = 3 και 5, τότε: $t0 = 1 (γιατί 3 < 5)

[Set on Less Than Unsigned (sltu): Αυτή η λειτουργία λειτουργεί το ίδιο με το slt, αλλά συγκρίνει unsigned ακέραιους αντί για signed ακέραιους.]

Βιβλιογραφία: MIPS Assembly Language using QtSpim Ed Jorgensen Version 1.0 January 2013