Branch and Jump Instructions part A

Branch Instructions in MIPS Assembly

Οι οδηγίες διακλάδωσης (Branch instructions) στη MIPS χρησιμοποιούνται για την αλλαγή της ροής ελέγχου ενός προγράμματος μεταβαίνοντας, υπό όρους, σε ένα διαφορετικό μέρος του κώδικα με βάση μια σύγκριση.

1. Διαφορετικοί τύποι Branch Instructions
   α. Equality Branches (beq, bne)
   β. Comparison Branches (bltz, blez, bgtz, bgez)
   γ. Set-and-Branch Combinations (slt with beq or bne)
   δ. Unconditional Jumps (j, jal, jr)

2. α. Equality Branches (beq, bne)

Branch if Equal (beq):

beq $rs, $rt, label ⇒ Εάν $rs == $rt, άλμα στην ετικέτα label. Διαφορετικά, η εκτέλεση συνεχίζεται κανονικά.

Παράδειγμα:

```
addi $t0, 5 # Φόρτωση 5 σε $t0
addi $t1, 5 # Φόρτωση 5 σε $t1
beq $t0, $t1, equal_case # Εάν $t0 == $t1, μεταβαίνει στο "equal_case" (αυτό εκτελείται εάν $t0 == $t1)
```

Branch if Not Equal (bne)

bne $rs, $rt, label ⇒ Εάν $rs != $rt, άλμα σε ετικέτα. Διαφορετικά, η εκτέλεση συνεχίζεται κανονικά.

Παράδειγμα:

```
addi $t0, 5 # Φόρτωση 5 σε $t0
addi $1, 10 # Φόρτωση 5 σε $t1
bne $t0, $t1, not_equal_case # Αν $t0 != $t1, άλμα στο not_equal_case:
                                        (Εκτελείται εάν $t0 != $t1)
```

β. Comparison Branches (bltz, blez, bgtz, bgez)

1. **bltz (Branch if Less Than Zero)**
   `bltz $rs, offset` (αν `$rs < 0` άλμα στην ετικέτα offset) `#Εκτελείται όταν το $rs είναι αρνητικό`

2. **blez (Branch if Less Than or Equal to Zero)**

   `blez $rs, offset` (αν `$rs <= 0` άλμα στην ετικέτα offset) `#Εκτελείται εάν $rs ≤ 0`

3. **bgtz (Branch if Greater Than Zero)**

   `bgtz $rs, offset` (αν `$rs > 0` άλμα στην ετικέτα offset) `#Εκτελείται όταν το $t0 είναι θετικό`

4. **bgez (Branch if Greater Than or Equal to Zero)**

   `bgez $rs, offset` (αν `$rs >= 0` άλμα στην ετικέτα offset) # Εκτελείται όταν $t0 ≥ 0

γ. Set-and-Branch Combinations (slt with beq or bne)

1. Διακλάδωση εάν $t0 > $t1

```
slt $t2, $t1, $t0                  # $t2 = 1 αν $t1 < $t0 (δηλαδή, $t0 > $t1)
bne $t2, $zero, greater_case       #Αν $t2 != 0, διακλάδωση greater_case
```

2. Διακλάδωση αν $t0 < $t1

```
slt $t2, $t0, $t1                  # $t2 = 1 εάν $t0 < $t1
bne $t2, $zero, less_case          # Αν $t2 != 0, διακλάδωση less_case
```

(αντίστοιχα παραδείγματα και με το beq)

Εφαρμογές

Looping Constructs: Χρησιμοποιείται για την υλοποίηση βρόχων όπως for, while και do-while.

Δηλώσεις υπό όρους: Εφαρμόζουμε τη λογική if και else.

Ροή ελέγχου λειτουργιών: Βοηθά στη λήψη αποφάσεων στο πλαίσιο κλήσεων συναρτήσεων.

Οι οδηγίες διακλάδωσης (Branch instructions) στο MIPS είναι απαραίτητες για τη ροή ελέγχου του προγράμματος. Η κατανόηση του τρόπου λειτουργίας τους επιτρέπει τον αποτελεσματικό προγραμματισμό.

Παραδείγματα

1)

```
.data
.text
.globl main
main:
addi $t0, 5 # Φόρτωση 5 σε $t0
addi $t1, 5 # Φόρτωση 5 σε $t1
beq $t0, $t1, label # If $t0 == $t1, διακλάδωση σε 'label'
add $t2, $t0, $t1 # Αυτή η εντολή παραλείπεται εάν εκτελεστεί  διακλάδωση
label:
sub $t3, $t0, $t1 # Η εκτέλεση συνεχίζεται εδώ εάν εκτελεστεί διακλάδωση
li $v0, 10 # Exit
syscall
```

2)

```
.data
msg1: .asciiz "The numbers are equal\n"
msg2: .asciiz "The first number is smaller\n"
msg3: .asciiz "The first number is greater\n"
.text
.globl main
main:
addi $t0, 10 # Load 10 into register $t0
addi $t1, 20 # Load 20 into register $t1

beq $t0, $t1, numbers_equal #beq: Branch if equal
slt $t2, $t0, $t1 # $t2 = 1 if $t0 < $t1, else $t2 = 0 # SLT: Set on less than
bne $t2, $zero, first_smaller # If not equal and not smaller, then $t0 is greater
```

```
li $v0, 4
la $a0, msg3
syscall
j exit

numbers_equal:
li $v0, 4
la $a0, msg1
syscall
j exit

first_smaller:
li $v0, 4
la $a0, msg2
syscall

exit:
li $v0, 10 # Exit
syscall
```

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