

## Assignment A P3 Lab 2 Pseudo-Direct Addressing

Elaborate on all the slides with the program execution results.

Kyu Lee, Ph. D.

### Lab 2: Pseudo-Direct Addressing

- Understanding the Pseudo-Direct Addressing
  - How to calculate the new effective address

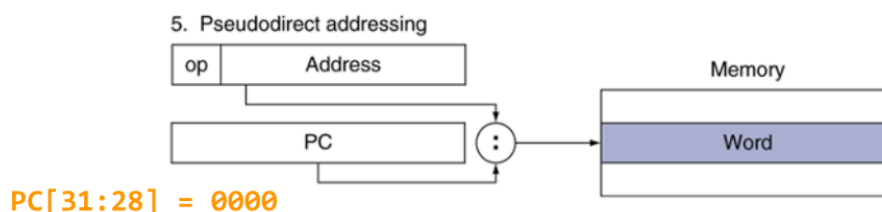


**Elaboration: Pseudo-Direct Addressing** explains how the **effective address** is calculated by combining the high 4 bits of the current program counter (PC[31:28]) with a 26-bit address from the instruction, and appending two trailing zero bits (00). This process results in a new 32-bit effective address used for branching or jumping in the program.

Kyu Lee, Ph. D.

### Lab 2: Pseudo-Direct Addressing

```
[00400038] 012a602a slt $t2, $9, $10          ; 32: slt      $t4, $t1, $t2  # i < N
[0040003c] 11800007 beq $t2, $0, 28 [exit-0x0040003c]; 33: beq      $t4, $0, exit
[00400040] 01a86820 add $t3, $t3, $8           ; 35: add      $t5, $t5, $t0
[00400044] 8dad0000 lw $t3, 0($t3)             ; 36: lw       $t5, 0($t5)
[00400048] 01ab5820 add $t1, $t3, $t1          ; 37: add      $t3, $t5, $t3
[0040004c] 21290001 addi $9, $9, 1             ; 38: addi     $t1, $t1, 1
[00400050] 00096880 sll $t3, $9, 2            ; 39: sll      $t5, $t1, 2
PC → [00400054] 0810000e j 0x00400038 [loop]    ; 40: j         loop
[00400058] 3c011001 lui $t1, 4097             ; 42: sw       $t3, sum
[0040005c] ac2b0010 sw $t1, 16($t1)
```



**Elaboration:** This showcases how the Pseudo-Direct Addressing works in MIPS assembly code. Highlighting the jump instruction (j) at memory address 0x00400054 we can see that it targets the label 'loop' at 0x00400038. The PC[31:28] segment is shown as 0000, and the instruction's 26-bit address (after decoding) is combined with this segment to compute the new effective address for the jump.

## Lab 2: Pseudo-Direct Addressing

```

[00400038] 012a602a slt $12, $9, $10
[0040003c] 11800007 beq $12, $0, 28 [exit-0x0040003c]
[00400040] 01a86820 add $13, $13, $8
[00400044] 8dad0000 lw $13, 0($13)
[00400048] 01ab5820 add $11, $13, $11
[0040004c] 21290001 addi $9, $9, 1
[00400050] 00096880 sll $13, $9, 2
PC → [00400054] 0810000e j 0x00400038 [loop]
[00400058] 3c011001 lui $1, 4097
[0040005c] ac2b0010 sw $11, 16($1)

```

```

0x0810 000E
0000 1000 0001 0000 0000 0000 0000 1110
000010 00 0001 0000 0000 0000 0000 1110
Op 2    Address 26 bits
PC: 0040 0054
PC[31:28] = 0000
Address 26 bits = 00 0001 0000 0000 0000 0000 1110
Address * 4 = 0000 0100 0000 0000 0000 0011 1000
New Effective Address = PC[31:28] : Address * 4
                     0000 0000 0100 0000 0000 0000 0011 1000
                     = 0x0040 0038

```

5. Pseudodirect addressing

**Elaboration:** The slide showcases a detailed breakdown of the calculations involved in Pseudo-Direct Addressing for the given example. We start with the instruction binary (0x0810000E), extract the 26-bit address (0000 1000 0000 0000 0000 0011 10), and multiply it by 4. By connecting this with the high 4 bits of the PC (0000), we get the final effective address 0x00400038, matching the target address for the jump instruction.