

Suponga que estamos trabajando con una arquitectura que posee 4 registros (\$R0, \$R1, \$R2 y \$R3) y la memoria es direccionada por palabras. El conjunto de instrucciones para esta arquitectura está conformado por las siguientes operaciones:

```

add   op1, op2           op1 = op1 + op2
sub   op1, op2           op1 = op1 - op2
move  op1, op2           op1 = op2

```

y los modos de direccionamiento son los siguientes:

```

Inmediato           #
Registro            $Ri
Directo             etiqueta o número
Indirecto           @
Desplazamiento      n($Ri)
Desplazamiento autoincremento n($Ri)+ (después de calcular la
                                         dirección del operando
                                         incrementa en 1 $Ri )

```

En un instante de tiempo los registros y la memoria presentan el siguiente estado:

\$R0	1
\$R1	0
\$R2	2
\$R3	0

Registros  
Del CPU

0	3
1	4
2	10
3	9
4	2
5	0
6	1
7	4
8	7
9	3
10	4
...	

Memoria

Dado el siguiente trozo de código y la tabla de símbolos, muestre cómo se modifican los registros y la memoria

```

move $R1 , #2
move $R0 , 2
move $R2 , B
add $R2 , 2($R1)
move C , @$R0
move $R0 , @4($R1)+

```

A	5
B	8
C	3

Tabla de  
símbolos

move \$R1 , #2

\$R1 referencia al registro 1 y #2 corresponde a un modo inmediato. La instrucción transfiere el valor 2 al registro

\$R0	1
\$R1	2
\$R2	2
\$R3	0

Registros  
Del CPU

0	3
1	4
2	10
3	9
4	2
5	0
6	1
7	4
8	7
9	3
10	4
...	

Memoria

move \$R0 , 2

\$R0 referencia al registro 0 y 2 corresponde a un modo directo. La instrucción transfiere el contenido de la dirección 2 de memoria al registro.

\$R0	10
\$R1	2
\$R2	2
\$R3	0

Registros  
Del CPU

0	3
1	4
2	10
3	9
4	2
5	0
6	1
7	4
8	7
9	3
10	4
...	

Memoria

move \$R2 , B

\$R2 referencia al registro 2 y B corresponde a un modo directo. Al revisar la tabla de símbolos obtenemos que la etiqueta B corresponde a la dirección 8 de memoria. La instrucción transfiere el contenido de la dirección de memoria 8 al registro.

\$R0	10
\$R1	2
\$R2	<u>7</u>
\$R3	0

Registros  
Del CPU

0	3
1	4
2	10
3	9
4	2
5	0
6	1
7	4
8	7
9	3
10	4
...	

Memoria

add \$R2 , 2(\$R1)

\$R2 referencia al registro 2 y 2(\$R1) corresponde a un modo desplazamiento. Para calcular la dirección del operando se toma el contenido del registro y se le suma el desplazamiento 2. En este caso la dirección efectiva del segundo operando es 4. Para la operación add se toma el contenido del registro 2 y el contenido de la dirección de memoria 4.

\$R0	10
\$R1	2
\$R2	<u>7</u>
\$R3	0

Registros  
Del CPU

0	3
1	4
2	10
3	9
4	2
5	0
6	1
7	4
8	7
9	3
10	4
...	

Memoria

move C, @\$R0

C referencia a la dirección 3 de memoria y @\$R0 corresponde a un modo indirecto registro. Para calcular la dirección del segundo operando se toma el contenido del registro 0 (10), y este valor es la dirección efectiva del operando. El contenido de la dirección de memoria 10 es transferido a la dirección 3

\$R0	10
\$R1	2
\$R2	9
\$R3	0

Registros  
Del CPU

0	3
1	4
2	10
3	<del>9</del> 4
4	2
5	0
6	1
7	4
8	7
9	3
10	4
...	

Memoria

move \$R0, @4(\$R1)+

\$R0 referencia al registro 0 y @4(\$R1) corresponde a un modo desplazamiento indirecto. Para calcular la dirección del segundo operando se toma el contenido del registro 1 (2), y a este valor se le suma el desplazamiento 4 para obtener una dirección de memoria (6). A esta dirección se le aplica el modo indirecto, es decir la dirección no contiene el operando sino la dirección que realmente contiene el operando. Por lo tanto el contenido de la dirección 1 de memoria es transferido al registro 0. Después de esta transferencia se incrementa el contenido del registro 1

\$R0	<del>10</del> 4
\$R1	<del>2</del> 3
\$R2	9
\$R3	0

Registros  
Del CPU

0	3
1	4
2	10
3	4
4	2
5	0
6	1
7	4
8	7
9	3
10	4
...	

