

CSI 333 – Programming at the Hardware-Software Interface

Example of MAL Code for accessing a 2-Dimensional Array

Handout 15.4

C Segment

```
int a[3][5]; int i,j;
for (i = 0; i < 3; i++)
    for (j = 0; j < 5; i++)
        a[i][j] = i*j;
```

We assume that the 2-dimensional array *a* is stored in *row-major* order. Variables *i* and *j* are stored in registers \$5 and \$6 respectively. Register \$7 is used for computing addresses and registers \$8 through \$12 are used for scratch work. It should be noted that the following code segment does not save/restore these registers.

MAL Segment

```
.data
a:      .word  0:15          #Space for a.

.text
move   $5, $0              #Set i = 0.
li     $10, 3              #$10 has the no. of rows of a.
li     $11, 5              #$11 has the no. of columns of a.
li     $12, 4              #$12 has 4 (no. of bytes per int.)

iloop: bge   $5, $10, end_iloop
move   $6, $0              #Set j = 0.
jloop: bge   $6, $11, end_jloop

#Calculate the address of a[i][j] in $7.

la     $7, a               #$7 has the starting addr. of a.
move   $8, $11
mul    $8, $8, $12         #$8 has 4 * no. of columns.
mul    $8, $8, $5          #$8 has 4 * no. of columns * i.
mul    $9, $6, $12         #$9 has 4*j.
add    $8, $8, $9          #$8 has the offset for a[i][j].
add    $7, $7, $8          #$7 has the addr. of a[i][j].

#Compute i*j and store it in a[i][j].

mul    $9, $5, $6          #$9 has i*j.
sw     $9, 0($7)           #Store i*j in a[i][j].
addi   $6, $6, 1          #j++
b      jloop

end_jloop: addi   $5, $5, 1      #i++
b      iloop

end_iloop:
```