## Assignment 2 Solution

<table>
<thead>
<tr>
<th>MIPS Assembly Code and HLL instructions</th>
<th># iterations</th>
<th># of machine ins per assembly ins</th>
<th>total machine ins</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>.verstamp 7 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.option pic2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.globl a1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.align 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.align 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a1:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 0:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 1:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 2:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 3:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 4:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 5:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 6:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 7:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 8:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 9:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 0:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 1:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 2:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 3:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 4:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 5:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 6:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 7:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 8:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 9:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 0:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 1:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 2:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 3:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.word 4:01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.globl a2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.align 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.align 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
.word 1:01
.word 131075 : 1
.word 262149 : 1
.word 393223 : 1
.word 524297 : 1
.word 1:01
.word 131075 : 1
.word 262149 : 1
.word 393223 : 1
.word 524297 : 1
word 1:01
.word 131075 : 1
.word 262149 : 1
.word 393223 : 1
.word 524297 : 1
# 15 {
.ent main 2

lw $4, 0($29)   1 1 1
addiu $5, $29, 4 1 1 1
addiu $6, $5, 4 1 1 1
sll $2, $4, 2 1 1 1
addu $6, $6, $2 1 1 1
jal 0x00400020 [main] 1 1 1

Inserted by SPIM

main:
.option O1
.set noreorder
.set reorder
subu $sp, 8 1 1 1 Subtract 8 from stack pointer
.frame $sp, 8, $31
.loc 2 18

# 16 int i, j;
# 17 for (i=0; i<5; i++)
# 18 for (j=0; j<5; j++)

sw $0, 4($sp)   1 1 1 Put 0 on Stack (i = 0)

sw $0, 0($sp)   5 1 5 Put 0 on Stack (j = 0)
# 20
```
a1[i][j] = a1[i][j] + i*j;
```

# 21
```
for (i=0;i<5;i++)
```

# 22
```
sw $0, 4($sp)
```

# 23
```
for (j=0;j<5;j++)
```

# 24
```
a2[i][j] = a2[i][j] + i*j;
```

```c
lw $14, 4($sp)    
lw $15, 0($sp)    
mul $24, $14, $15
mul $25, $14, 20
mul $8, $15, 4   
addu $9, $25, $8
la $10, a1       
add $11, $9, $10
lw $12, ($11)    
add $13, $12, $24
addu $14, $25, $8
addu $15, $14, $10
sw $13, ($15)    
lw $9, 0($sp)     
addu $11, $9, 1   
sw $11, 0($sp)    
add $12, 4($sp)   
addu $24, $12, 1  
sw $24, 4($sp)    
blt $11, 5, $32   
```

```c
lw $12, 4($sp)     
add $14, $25, $8   
mul $10, $25, 10  
mul $13, $8, 2    
add $15, $10, $13 
la $9, a2         
add $11, $15, $9  
lw $12, 0($11)    
add $24, $12, $14 
add $25, $10, $13 
add $8, $25, $9   
sh $24, 0($sp)    
```

```
```
lw $15, 0($sp)  25  1  25  Get j from stack
addu $11, $15, 1  25  1  25  j++
sw $11, 0($sp)  25  1  25  Put new value of j on stack
blt $11, 5, $35  25  2  50  branch to $35 if j < 5
lw $12, 4($sp)  5  1  5  Get i from stack
addu $14, $12, 1  5  1  5  i++
sw $14, 4($sp)  5  1  5  put new value of i on stack
blt $14, 5, $34  5  2  10  branch to $34 if i < 5

move $2, $0  1  1  1
.livereg 0x2000FF0E,0x00000FFF  1  1  1
addu $sp, 8  1  1  1
j $31  1  1  1
.end main

Inserted by
ori $2, $0, 10  1  1  1
syscall  1  1  1

Assembly Instruction Count -> 910  1249  <- Machine Instruction Count
Ratio (machine/assembly) -> 1.37