Ayman Hajja. Wed Jul 26 2017 14:18:37 GMT-0400 (EDT)

****

1) Translate the following C code to MIPS assembly code.

```
int arr[20];
void main()
{
int counter = 0;
  int s0;
  int s1;
  for(s0 = 0; s0 < 5; s0++)
    for (s1 = 0; s1 < 4; s1++)
    {
      arr[counter] = s0 + s1;
counter = counter + 1;
    }
}
.data
arr: .space 80
.text
main:
add $s2, $zero, $zero     # int counter = 0
add $s0, $zero, $zero
OUTER:
slti $t4, $s0, 5
beq $t4, $zero, EXITOUTER
add $s1, $zero, $zero
INNER:
slti $t5, $s1, 4
beq $t5, $zero, EXITINNER
add $t6, $s0, $s1
sll $t7, $s2, 2       # Multiply by 4
sw $t6, arr($t7)
addi $s2, $s2, 1    # counter = counter + 1
addi $s1, $s1, 1
j INNER
EXITINNER:
addi $s0, $s0, 1
j OUTER
EXITOUTER:
addi $v0, $zero, 10
syscall
```

****

2) Provide a set of pure MIPS instructions that may be used to implement the following pseudo-instruction:

```
not $t1, $t2 # bit-wise invert
nor $t1, $t2, $t2
```

****

3) Translate the following C code to MIPS assembly code.

```
void main()
{
int counter = 20;
int loc_arr[16];

  while(counter >= 5)
  {
    loc_arr[counter - 5] = 50;
    counter = counter - 1;
  }
}
.text
main:
addi $s0, $zero, 20   # int counter = 20
addi $sp, $sp, -64  # Allocate space for 16 ints
addi $t3, $zero, 50
BEFORE:
slti $t1, $s0, 5
bne $t1, $zero, EXIT
addi $t2, $s0, -5
sll $t2, $t2, 2
add $t2, $sp, $t2
sw $t3, 0($t2)
addi $s0, $s0, -1
j BEFORE
EXIT:
addi $v0, $zero, 10
syscall
```

****

4) Assume $t0 holds the value 0x00101000. What is the value of $t2 after the following instructions?

```
slt $t2, $0, $t0
bne $t2, $0, ELSE
j DONE
ELSE:
addi $t2, $t2, 2
DONE:
```
3

****

5) Translate the following C code to MIPS assembly code.

```
int counter = 0;
```

```
void change_global(int value)
{
counter = counter + value;
}

void main()
{
change_global(5);
change_global(10);
}
.data
counter: .word 0
.text
main:
addi $a0, $zero, 5
jal change_global
addi $a0, $zero, 10
jal change_global
addi $v0, $zero, 10
syscall
change_global:
lw $t0, counter($zero)
add $t0, $t0, $a0
sw $t0, counter($zero)
jr $ra
```

****