BPOS: Excercises for week 7

student: Dimitri Tabatadze

May 6, 2023

Solutions

1. The code in question:

```
int b;
int y;

int main() {
    b = 5;
    y = 11;
    if b > 1 {
        b = f1(b, b - 1)
    };
    return 0
};

int f1(int y, int z) {
    while b > 0 {
        if y > 12 {
            b = (y - 1) * f1(y - 1, y)
        } else {
            if y > 9 {
                y = y - 1;
                b = 5
            } else {
                b = 5;
                y = y - 1
            }
        }
    };
    y = y - 1;
    return b
}
```

   - $c'.pr =$ `b=5;`$ft(f).body$`;return 0`
   - $c'.pr =$ $ft(f).body$`;return 0`
   - $c'.pr =$ `return b;return 0`

2. The expression to translate: `z + x[z]`

$$displ(\texttt{z}, \$gm) = size(vec) = (6 \cdot size(int))_{32} = (6 \cdot 4)_{32} = 24_{32}$$
$$displ(\texttt{x}, \$gm) = 0_{32}$$

The register `bpt`, where the pointer to the start of the region where global memory is contained, is the register number 28 which is called `$gp` in MIPS.

The generated MIPS code:

- Load `x[z]` into `$t0`

```
addi $t0 bpt displ(x, $gm)
deref($t0)
addi $t1 bpt displ(z, $gm)
deref($t1)
addi 23 0 size(int)
mul($t1, $t1, 23)
add $t0 $t0 $t1
deref($t0)
```

- Load `z` into `$t1`

```
addi $t1 bpt displ(z, $gm)
deref($t1)
```

- perform addition

```
add $t0 $t0 $t1
```

After expanding the macros:

```
addi $t0 $gp 0
lw $t0 $t0 0
addi $t1 $gp 24
lw $t1 $t1 0
addi $s7 $zero 4
mul $t1 $t1 $s7
add $t0 $t0 $t1
lw $t0 $t0 0
addi $t1 $gp 24
lw $t1 $t1 0
add $t0 $t0 $t1
```

3. The expression to translate: `while z >0 { z = z - 2 }`

$$displ(\texttt{z}, \$gm) = size(vec) = (6 \cdot size(int))_{32} = (6 \cdot 4)_{32} = 24_{32}$$

The register `bpt`, where the pointer to the start of the region where global memory is contained, is the register number 28 which is called `$gp` in MIPS.

The generated MIPS code:

- `code(n1)`

```
addi $t0 bpt displ(z, $gm)
deref($t0)
addi $t1 0 0
slt $t0 $t0 $t1
```

- `beqz j |code(n3)|+2`

```
beqz $t0 5+2
```

- `code(n3)`

```
addi $t1 bpt displ(z, $gm)
deref($t1)
addi $t2 0 2
sub $t1 $t1 $t2
sw $t1 $t2 0
```

- `blez 0 -(|code(n1)|+|code(n3)|+1)`

```
blez 0 -(4+5+1)
```

After expanding the macros:

```
addi $t0 28 24
lw $t0 $t0 0
addi $t1 $zero 0
slt $t0 $t0 $t1
beqz $t0 7
addi $t1 28 24
lw $t1 $t1 0
addi $t2 $zero 2
sub $t1 $t1 $t2
sw $t1 $t2 0
blez $zero -10
```