IPS - Assignment 3

Mikkel Willén, bmq419

2. maj 2023

Indhold

1 Task

1.1 Intermediate code

```
1        t_0  =  v_0
2        t_1  =  v_1
3
4  Label L1
5        t_2  =  t_0
6        t_3  =  t_1
7        t_4  =  0
8        if  t_3  ==  t_4  then  endlab  else  L2
9
10 Label L2
11       t_5  =  t_2
12       t_6  =  t_3
13       t_7  =  t_5/t_6
14       t_8  =  1
15       if  t_8  <  t_7  then  L3  else  endlab
16
17 Label L3
18       if  t_3  <  t_2  then  L4  else  L5
19
20 Label L4
21       t_0  =  t_2  -  t_3
22       GOTO  L1
23
24 Label L5
25       t_1  =  t_3  -  t_2
26       GOTO  L1
27
28 Label endlab
```

1.2 MIPS code

```
1        add      t_0 ,  R_0 ,  v_0
2        add      t_1 ,  R_0 ,  v_1
3
4  Label L1:
5        add      t_2 ,  R_0 ,  t_0
6        add      t_3 ,  R_0 ,  t_1
7        add      t_4 ,  R_0 ,  0
8        beq      t_3 ,  t_4 ,  endlab
9
10 Label L2:
11       add      t_5 ,  R_0 ,  t_2
12       add      t_6 ,  R_0 ,  t_3
13       div      t_7 ,  t_5 ,  t_6
14       addi     t_8 ,  R_0 ,  1
15       slt      t_9 ,  t_8 ,  t_7
16       beq      t_9 ,  R_0 ,  endlab
17
```

```
18  Label L3:
19      slt        t9, t3, t2
20      beq        t9, R0, L5
21
22  Label L4:
23
24      sub        t0, t2, t3
25      j          L1
26
27  Label L5:
28      sub        t1, t3, t2
29      j          L1
30
31  Label endlab:
```

2   Task

pattern	replacement	
z:= x >= y	slt	r_d, r_x, r_y
	xori	r_z, r_d, 1
w:= !z	mul	r_w, r_z, r_z
	slti	r_w, r_w, 1
z:= x >= y	slt	r_w, r_x, r_y
w:= !z		

3   Task

3.1   C-like

```
1  scan(fun myop, bool ne, bool* x) = {
2      int len = length(x)
3      bool* y = malloc(len * 4)
4      int i = 0
5      while (i < len) {
6          bool temp =  myop(ne, x[i])
7          y[i] = temp
8          i = i + 1
9      }
10 }
```

3.2   MIPS code

```
1      lw         R_len, 0(R_x)
2      move       R_y, R_HP
3      sll        R_temp, R_len, 2
4      addi       R_temp, R_temp, 4
5      add        R_HP, R_HP, R_temp
6      sw         R_len, 0(R_y)
7      addi       R_ix, R_x, 4
8      addi       R_iy, R_y, 4
9      move       R_i, $0
```

```
10  loop_begin:
11      sub       R_temp, R_i, R_len
12      bgez      R_temp, loop_end
13      lw        R_temp, 0(R_ix)
14      addi      R_ix, R_ix, 4
15      move      R_clen, $8
16      move      R_c, R_HP
17      add       R_HP, R_HP, R_clen
18      sw        R_clen, 0(R_c)
19      sw        R_ne, 4(R_c)
20      sw        R_temp, 8(R_c)
21      call      myop, R_c, R_temp
22      sw        R_temp, 0(R_iy)
23      addi      R_iy, R_iy, 4
24      addi      R_i, R_i, 1
25      j         loop_begin
26  loop_end:
```