5.10 Conserving registers

Registers are limited, so computations should conserve registers, lest the computation require more registers than exist. If register is not read later, the register can be reused by writing another value.

PARTICIPATION ACTIVITY 5.10.1: Conserving registers.

Start 2x speed

```
u = v + w + x + y + z;
```

```
tmp1 = v + w;
tmp2 = tmp1 + x;
tmp3 = tmp2 + y;
u = tmp3 + z;
```

```
add $t5, $t0, $t1
add $t6, $t5, $t2
add $t7, $t6, $t3
add $t8, $t7, $t4
```

Assume $t0-$t4 are occupied; only $t5 and $t6 are available

```
add $t5, $t0, $t1
add $t5, $t5, $t2
add $t5, $t5, $t3
add $t5, $t5, $t4
```

Say v..z are 50, 40, 30, 20, and 10

$t5 90 120 140 150

PARTICIPATION ACTIVITY 5.10.2: Conserving registers.

Assume no temporary values (tmp1, tmp2) are used by later instructions. Rewrite the instructions to reuse $t4, to conserve registers.

1)

```
tmp1 = x + y;
w = tmp1 + z;

add $t4, $t0, $t1
add $t5, $t4, $t2
sw $t5, 0($t3)

add $t4, $t0, $t1
```

```
sw $t4, 0($t3)
```

Check Show answer

2)

```
tmp1 = w + x;
tmp2 = tmp1 + y;
w = tmp2 + 9;

add $t4, $t0, $t1
add $t5, $t4, $t2
addi $t6, $t5, 9
sw $t6, 0($t3)

add $t4, $t0, $t1
```

```
sw $t4, 0($t3)
```

Check Show answer

3)

```
w = (w + x) * (y + z)

add $t4, $t0, $t1 # tmp1 = w + x
add $t5, $t2, $t3 # tmp2 = y + z
mul $t6, $t4, $t5 # tmp3 = tmp1 * tmp2
sw $t6, ...
```

For the above code, 3 registers are used to hold temporary values. That number can be reduced to _____ .
Type 3, 2, or 1.

Check Show answer

CHALLENGE ACTIVITY 5.10.1: Conserving registers.

Start

1

Rewrite the instructions to reuse $t4, to conserve registers.

2

```
tmp1 = w + y;
x = z - tmp1;
```

3

```
add $t4 , $t0 , $t1
sub $t5 , $t2 , $t4
sw  $t5 , 0( $t3 )
```

Registers			Data memory		
$t0	2	w	5008	0	x
$t1	7	y			
$t2	11	z			
$t3	5008				
$t4	0				
$t5	0				

1 2 3

Check Next

Provide feedback on this section