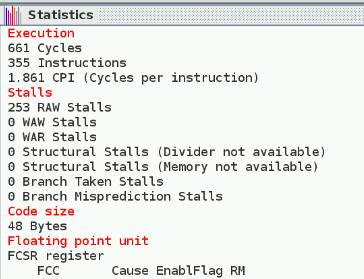
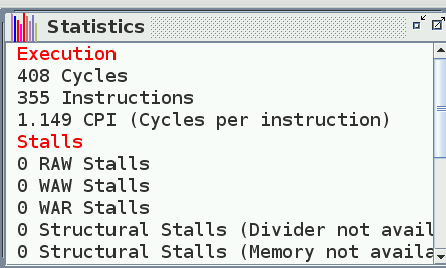
3 Tasks

# 3.1 Task 1: Evaluation of example 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction no. source** | **Instruction no. target** | **Register** | **Type** |
| 0x0 | 0x4 | R1 | RAW |
| 0x10 | 0x8 | R3 | RAW |
| 0x14 | 0x18 | R4 | RAW |
| 0x18 | 0x1C | R4 | RAW |
| 0x20 | 0x28 | R3 | RAW |

No forwarding

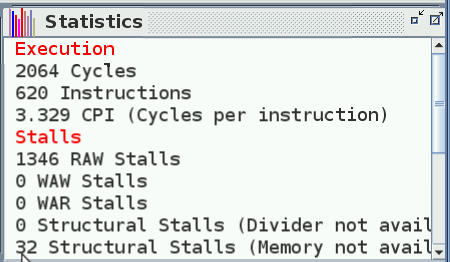
Forwarding



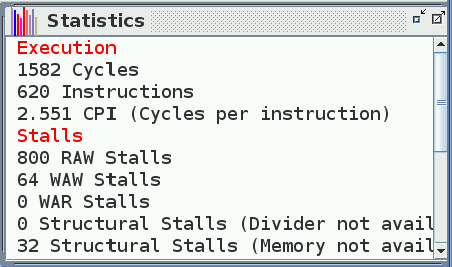
# 3.2 Task 1: Evaluation of example 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction no. source** | **Instruction no. target** | **Register** | **Type** |
| 0x14 | 0x18 | R4 | RAW |
| 0x18 | 0x1C | F6 | RAW |
| 0x1C | 0x20 | F6 | RAW |
| 0x20 | 0x24 | F8 | RAW |
| 0x24 | 0x28 | F8 | RAW |
| 0x2C | 0x30 | F10 | RAW |
| 0x3C | 0x40 | R4 | RAW |
| 0x54 | 0X5C | F2 | RAW |
| 0x58 | 0x5C | F4 | RAW |
| 0x5C | 0x60 | F6 | RAW |
| 0x70 | - | - | Structural stall |
| 0x6C | 0x70 | R4 | RAW |

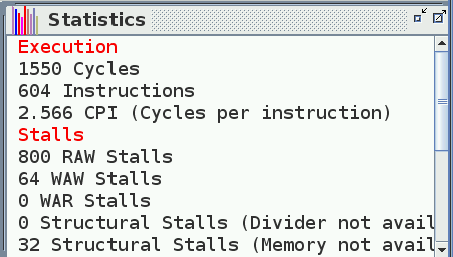
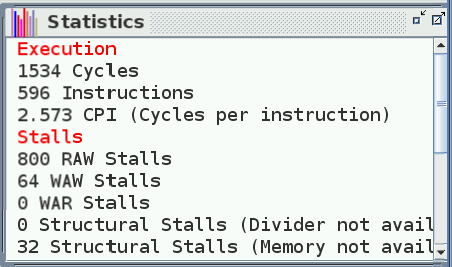
No forwarding



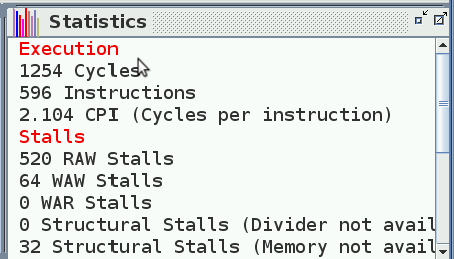
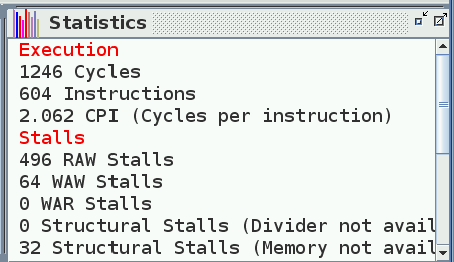
Forwarding



# 3.3 Task 3: Loop unrolling, first loop

We get a 32-cycle improvement and a 48-cycle improvement when applying only loop unrolling with factors 2 and 4, respectively.

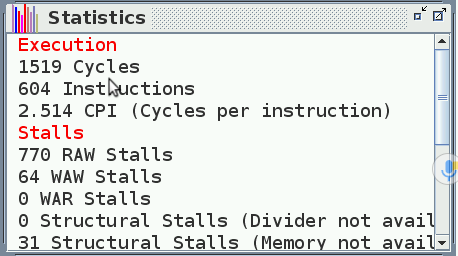
 

When scheduling the loop, we save 296 on the first unrolling, and 288 on the second one. The code was rearranged in the following way:

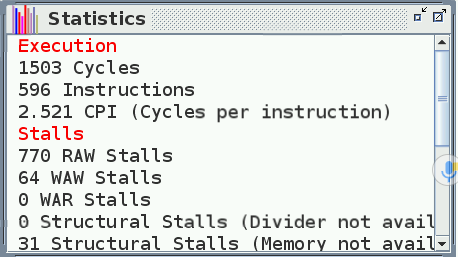
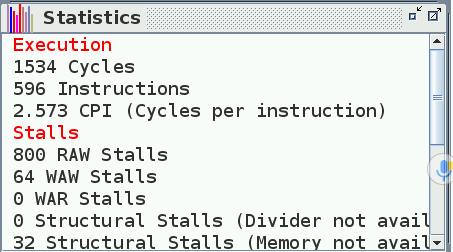
*dmtc1 R4, F6  
cvt.d.l F6, F6  
mul.d F8, F2, F6  
mul.d F8, F8, F8  
sdc1 F8, 0(R1)  
mul.d F10, F4, F6  
sdc1 F10, 0(R2)  
daddi R1, R1, 8  
daddi R2, R2, 8  
daddi R4, R4, 8*

*dmtc1 R4, F6  
cvt.d.l F6, F6  
mul.d F8, F2, F6  
mul.d F10, F4, F6  
mul.d F8, F8, F8  
daddi R4, R4, 8  
daddi R1, R1, 8  
daddi R2, R2, 8  
sdc1 F10, 0(R2)  
sdc1 F8, 0(R1)*

# 3.4 Task 4: Loop unrolling, second loop



We get a 32-cycle improvement and a 63-cycle improvement when applying loop unrolling with factors 2 and then scheduling.



We get a 48-cycle improvement and a 79-cycle improvement when applying loop unrolling with factors 4 and then scheduling.

*ldc1 F2, 0(R1)*

*ldc1 F4, 0(R2)*

*mul.d F6, F2, F4*

*add.d F8, F8, F6*

*daddi R1, R1, 8*

*daddi R2, R2, 8*

*daddi R4, R4, 8*

*ldc1 F2, 0(R1)*

*ldc1 F4, 0(R2)*

*daddi R1, R1, 8*

*daddi R2, R2, 8*

*daddi R4, R4, 8*

*mul.d F6, F2, F4*

*add.d F8, F8, F6*

# 3.5 Task 5: Loop unrolling

