|  |  |  |
| --- | --- | --- |
| PROF. AKKARY | DEPT. OF ELECTRICAL AND COMPUTER ENGINEERING |  |
|  | AMERICAN UNIVERSITY OF BEIRUT |  |
|  | **EECE 421 – COMPUTER Architecture** |  |
|  | **Quiz 3 – Fall 2012** |  |
|  |  |  |
| **NAME**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |
| **ID**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |

**INSTRUCTIONS:**

* **The duration of the exam is 90 minutes. No time extension.**
* **The exam is closed-book/closed-notes.**
* **Using Cell phones is not allowed in the examination room.**
* **Write your name and ID. NumBer in the space provided above.**
* **Circle only one answer.**
* **READ THE QUESTIONS CAREFULLY BEFORE ANSWERING.**
* **in some questions, more than one choice may be a valid answer. Circle the best choice you think is the most appropriate answer to the question.**
* **ALL QUESTIONS ARE EQUALLY WEIGHTED.**
* **There is no penalty for wrong answers.**
* **Use the back pages for scratch if needed**
* **Check that you have a total of 5 pages.**
* **No questions are allowed.**
* **You cannot leave the exam room for any reason until you complete the exam.**

Consider the following loop that adds two single precision floating point arrays and the writes the result in both arrays. Assume 5 stage MIPS pipeline with complete bypass, 3 cycles floating point Add execution unit, and BEQ execution of the condition and target in the decode stage. Assume that the pipeline **does not** perform delayed branch execution, (i.e., BEQ jumps to target immediately without allowing the pipeline to execute the next instruction after the branch in program memory).

Loop: Ld F0, 0(R1)

Ld F1, 0(R2)

Add F2, F0, F1

St F2, 0(R1)

St F2, 0(R2)

Add R1, R1, #4

Add R2, R2, #4

BNE R2, R4, Loop

1. Without unrolling or software scheduling, how many cycles per iteration it takes to execute the loop. Round up if the answer is a fraction:
	1. 7 cycles
	2. 8 cycles
	3. 9 cycles
	4. 10 cycles
	5. 11 cycles
	6. 12 cycles
	7. 13 cycles
	8. None of the above

Loop: Ld F0, 0(R1)

Ld F1, 0(R2)

Stall

Add F2, F0, F1

Stall

Stall

St F2, 0(R1)

St F2, 0(R2)

Add R1, R1, #4

Add R2, R2, #4

Stall

BNE R2, R4, Loop

Stall

1. If the compiler schedules the loop without loop unrolling, how many cycles per iteration it takes to execute the loop. Round up if the answer is a fraction:
	1. 7 cycles
	2. 8 cycles
	3. 9 cycles
	4. 10 cycles
	5. 11 cycles
	6. 12 cycles
	7. 13 cycles
	8. None of the above

Loop: Ld F0, 0(R1)

Ld F1, 0(R2)

Add R1, R1, #4

Add F2, F0, F1

Add R2, R2, #4

Stall

St F2, -4(R1)

St F2, -4(R2)

BNE R2, R4, Loop

Stall

1. If the compiler unrolls the loop twice (i.e. new loop iteration consists of two old loop iterations), how many cycles per iteration it takes to execute the loop. Round up if the answer is a fraction:
	1. 7 cycles
	2. 8 cycles
	3. 9 cycles
	4. 10 cycles
	5. 11 cycles
	6. 12 cycles
	7. 13 cycles
	8. None of the above

Loop: Ld F0, 0(R1)

Ld F1, 0(R2)

Stall

Add F2, F0, F1

Stall

Stall

St F2, 0(R1)

St F2, 0(R2)

Ld F0, 4(R1)

Ld F1, 4(R2)

Stall

Add F2, F0, F1

Stall

Stall

St F2, 4(R1)

St F2, 4(R2)

Add R1, R1, #8

Add R2, R2, #8

Stall

BNE R2, R4, Loop

Stall

1. If the compiler unrolls the loop twice and then performs instruction scheduling, how many cycles per iteration it takes to execute the loop. Round up if the answer is a fraction:
	1. 7 cycles
	2. 8 cycles
	3. 9 cycles
	4. 10 cycles
	5. 11 cycles
	6. 12 cycles
	7. 13 cycles
	8. None of the above

Loop: Ld F0, 0(R1)

Ld F1, 0(R2)

Ld F2, 4(R1)

Add F4, F0, F1

Ld F3, 4(R2)

Add R1, R1, #8

Add R2, R2, #8

Add F5, F2, F3

St F4, -8(R1)

St F4, -8(R2)

St F5, -4(R1)

St F5, -4(R2)

BNE R2, R4, Loop

Stall

1. The compiler unrolls the loop twice and schedules the instructions on a 2-wide superscalar with one floating point execution unit, and one load/store execution port.

Complete the next statement:

Cycles per iterations to execute the loop = 5.5

Loop: Ld F0, 0(R1) Add R1, R1, #8

Ld F1, 0(R2) Add R2, R2, #8

Ld F2, -4(R1)

Ld F3, -4(R2) Add F4, F0, F1

Stall

Add F5, F2, F3

St F4, -8(R1)

St F4, -8(R2)

St F5, -4(R1)

St F5, -4(R2) BNE R2, R4, Loop

Stall

1. The compiler unrolls the loop twice and schedule the instructions to execute on a 3-wide VLIW processor that can execute a memory op, a floating op or an integer/branch op in the same cycle.

Complete the next statement:

Number of NOPs in the compiled loop = 20

Loop: Ld F0, 0(R1) Add R1, R1, #8 NOP

Ld F1, 0(R2) Add R2, R2, #8 NOP

Ld F2, -4(R1) NOP NOP

Ld F3, -4(R2) NOP Add F4, F0, F1

NOP NOP NOP

NOP NOP Add F5, F2, F3

St F4, -8(R1) NOP NOP

St F4, -8(R2) NOP NOP

St F5, -4(R1) NOP NOP

St F5, -4(R2) BNE R2, R4, Loop NOP

NOP NOP NOP

1. Assume the VLIW processor in the problem above uses stop bits.

Complete the next statement:

Number of operations in the compiled loop = 13

1. An Itanium EPIC compiler uses data speculation to move the load instruction and the instructions that uses the load data in the following code above the store.

Store R1, (R2)

Ld R3, (R4)

Add R5, R3, #1

Complete the next statement:

The compiled code segment will have 7 instructions.

Scratch Page

Scratch Page