***Use the following code fragment:***

|  |  |
| --- | --- |
| ***loop: LD***  ***DADDI***  ***SD***  ***DADDI DSUB***  ***BNEZ*** | ***R1,0(R2)R1,R1,1 0(R2),R1 R2,R2,4 R4,R3,R2 R4,loop*** |

***Assume that the initial value of R3 is R2+396. Throughout this exercise use the classic RISC five-stage integer pipeline in H&P. Specifically, assume that: 1) branches are resolved in the second stage of the pipeline; 2) there are separate instruction and data memories; 3) all memory accesses take 1 clock cycle.***

***a. Show the timing of this instruction sequence for the RISC pipeline without any forwarding***

***or bypassing hardware but assuming a register read and a write in the same clock cycle***

***\forwards" through the register file. Assume that the branch is handled by flushing the***

***pipeline. If all memory references take 1 cycle, how many cycles does this loop take to***

***execute?***

1. Loop Structure:

loop: LD R1,0(R2) # 1

DADDI R1,R1,1 # 2

SD 0(R2),R1 # 3

DADDI R2,R2,4 # 4

DSUB R4,R3,R2 # 5

BNEZ R4,loop # 6

* + Iterations: 99 (since R4 starts at 396, decrements by 4 each loop).
  + Branch Penalty: 1 cycle per taken branch (99 times).

1. Data Hazards and Stalls:
   * LD → DADDI R1: RAW hazard on R1.
     + LD writes R1 in WB (cycle 5).
     + DADDI R1 needs R1 in ID (cycle 2).
     + Stalls: 3 cycles (ID of DADDI stalls until LD’s WB).
   * DADDI R1 → SD: RAW hazard on R1.
     + DADDI R1 writes R1 in WB (cycle 8).
     + SD reads R1 in EX (cycle 9).
     + No stalls (value available after WB).
   * DADDI R2 → DSUB: RAW hazard on R2.
     + DADDI R2 writes R2 in WB (cycle 13).
     + DSUB reads R2 in ID (cycle 14).
     + No stalls.
2. Pipeline Timing (One Iteration):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| LD | IF | ID | EX | MEM | WB |  |  |  |  |  |  |  |  |  |  |
| DADDI R1 |  | IF | ID | S | S | S | EX | MEM | WB |  |  |  |  |  |  |
| SD |  |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |  |
| DADDI R2 |  |  |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |  |
| DSUB |  |  |  |  |  |  |  | IF | ID | EX | MEM | WB |  |  |  |
| BNEZ |  |  |  |  |  |  |  |  | IF | ID | EX | MEM | WB |  |  |

* + Branch Penalty: 1 cycle added after BNEZ (total per iteration = 14 cycles).

1. Total Cycles:
   * 99 iterations: 99×14=1386cycles.
   * Final iteration (not taken branch): 13 cycles (no penalty).
   * Total: 1386+13=1399​ cycles

***b. Show the timing of this instruction sequence for the RISC pipeline with normal forwarding***

***and bypassing hardware. Assume that the branch is handled by predicting it as not taken.***

***If all memory references take 1 cycle, how many cycles does this loop take to execute?***

1. Forwarding Eliminates Stalls:
   * LD → DADDI R1: Forward from MEM to EX (1 stall).
   * DADDI R1 → SD: Forward from EX to EX (no stalls).
   * DADDI R2 → DSUB: Forward from EX to ID (no stalls).
2. Pipeline Timing (One Iteration):

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instruction | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| LD | IF | ID | EX | MEM | WB |  |  |  |  |
| DADDI R1 |  | IF | ID | EX | MEM | WB |  |  |  |
| SD |  |  | IF | ID | EX | MEM | WB |  |  |
| DADDI R2 |  |  |  | IF | ID | EX | MEM | WB |  |
| DSUB |  |  |  |  | IF | ID | EX | MEM | WB |
| BNEZ |  |  |  |  |  | IF | ID | EX | MEM | WB |

* + Branch Misprediction Penalty: 2 cycles (pipeline flushes 2 instructions).
  + Cycles per Iteration: 10 cycles (including 2 misprediction penalties).

1. Total Cycles:
   * 99 iterations: 99×10=990cycles.
   * Final iteration (correct prediction): 8 cycles.
   * Total: 990+8=998​ cycles.

***c. Assume the RISC pipeline with a single-cycle delayed branch and normal forwarding and***

***bypassing hardware. Schedule the instructions in the loop including the branch delay slot.***

***You may reorder instructions and modify the individual instructions operands, but do not***

***undertake other loop transformations that change the number or opcode of the instructions***

***in the loops. Show a pipeline timing diagram and compute the number of cycles needed to***

***execute the entire loop***

1. Branch Delay Slot: Fill with useful instruction. Reorder instructions:

loop: LD R1,0(R2) # 1

DADDI R2,R2,4 # 4 (moved up)

DADDI R1,R1,1 # 2

SD -4(R2),R1 # 3 (offset adjusted)

DSUB R4,R3,R2 # 5

BNEZ R4,loop # 6

NOP # Delay slot (eliminated)

1. Pipeline Timing (No Stalls):
   * Delay Slot Filled: DSUB moved into delay slot.
   * Cycles per Iteration: 6 cycles (no stalls due to forwarding and scheduling).
2. Total Cycles:
   * 99 iterations: 99×6=594cycles.
   * Final iteration: 6 cycles.
   * Total: 594+6=600​ cycles.