

CMSC611: Advanced Computer Architecture

Homework 3

Question 1:

(50 points)

Consider the following loop which implements a vector operation $Y = a \times X + Y$:

```

LOOP: L.D      F2,    0(R1)
      MUL.D    F4,    F2,    F0
      L.D      F6,    0(R2)
      ADD.D    F6,    F4,    F6
      S.D      0(R2), F6
      DADDIU   R1,    R1,    #8
      DADDIU   R2,    R2,    #8
      DSUBIU   R3,    R3,    #16
      BNEZ    R3,    LOOP
    
```

We have the following assumptions for this question:

1. Forwarding logic
2. Delayed branching
3. Branch is resolved in the ID stage
4. Floating-point units are fully pipelined
5. Instruction fetch and memory access can occur simultaneously without any structural hazard

Latencies for different function units are shown as:

Producing Instruction	Instruction using results	Latency in clock cycles
Integer ALU	Integer ALU	0
FP Add	FP ALU op	3
FP Multiply	FP ALU op	6
Load	FP ALU op	1
FP Add	Store	2
FP Multiply	Store	5

- a) Draw the FP multi-cycle pipeline chart using the latency table and show possible stalls for one loop execution. How many clock cycles does one loop iteration take?
- b) Schedule the loop by reordering instructions to reduce the execution time. Repeat part a) for your scheduled code.
- c) Assume the initial value of R3 is 64. Unroll the loop and schedule it. Show your unrolled and scheduled codes. Which code performs better on clock cycles?

Question 2:

(50 points)

Consider Tomasulo's algorithm with the following settings:

Functional Unit	Number of Reservation Stations	Number of Functional Units	Execution Cycles in the EX stage
Integer and Load/Store	2	2	2
FP Adder	2	2	4
FP Multiplier	2	1	8
FP Divider	2	1	30

For the following code:

```

L.D      F0,    8(R1)
L.D      F2,    8(R2)
ADD.D    F4,    F2,    F4
MUL.D    F8,    F6,    F4
SUB.D    F6,    F10,   F0
DIV.D    F12,   F6,    F2
S.D      16(R3), F12
ADD.D    F12,   F8,    F2
S.D      16(R4), F12
DADDIU   R1,    R1,    #8
DADDIU   R2,    R2,    #8

```

- Show the instruction status table after using the Tomasulo algorithm to execute the code. Use the format of the instruction status table presented in the lecture slides "09-tomasulo" and show all cycles occupied in the execution stage using a range number, e.g., if one instruction in the execution stage takes the 4th, 5th, 6th, and 7th cycles, show "4-7" in the execution stage. Please notice that you only need to show the final result of the instruction status table after the code execution is completed rather than show the table for each execution cycle.
- Show the reservation station and register result status tables during the execution of the code on the 4th, 6th, and 13rd cycles. Use the format of the reservation station and register result status tables presented in the lecture slides "09-tomasulo".