Problem M3.8: Multithreading [?? Hours]

Problem M3.8.A

Since there is no penalty for conditional branches, instructions take one cycle to execute unless there is a dependency problem. The following table summarizes the execution time for each instruction. From the table, the loop takes **104 cycles** to execute.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Start Cycle</th>
<th>End Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>LW R3, 0(R1)</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>LW R4, 4(R1)</td>
<td>2</td>
<td>101</td>
</tr>
<tr>
<td>SEQ R3, R3, R2</td>
<td>101</td>
<td>101</td>
</tr>
<tr>
<td>BNEZ R3, End</td>
<td>102</td>
<td>102</td>
</tr>
<tr>
<td>ADD R1, R0, R4</td>
<td>103</td>
<td>103</td>
</tr>
<tr>
<td>BNEZ R1, Loop</td>
<td>104</td>
<td>104</td>
</tr>
</tbody>
</table>

Problem M3.8.B

If we have N threads and the first load executes in cycle 1, **SEQ**, which depends on the load, executes in cycle $2N + 1$. To fully utilize the processor, we need to hide the 100-cycle memory latency, $2N + 1 \geq 101$. The minimum number of thread needed is **50**.

Problem M3.8.C

<table>
<thead>
<tr>
<th>Throughput</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Better</td>
<td>✓</td>
</tr>
<tr>
<td>Same</td>
<td></td>
</tr>
<tr>
<td>Worse</td>
<td>✓</td>
</tr>
</tbody>
</table>

Problem M3.8.D

In steady state, each thread can execute 6 instructions (**SEQ**, **BNEZ**, **ADD**, **BNEZ**, **LW**, **LW**). Therefore, to hide 99 cycles between the second **LW** and **SEQ**, a processor needs $\left\lceil \frac{99}{6} \right\rceil + 1 = 18$ threads.
Problem M3.9: Multithreaded architectures [?? Hours]

Problem M3.9.A

4, since the largest latency for any instruction is 4.

Problem M3.9.B

$\frac{2}{12} = 0.17$ flops/cycle, on average we complete a loop every 12 cycles

Problem M3.9.C

Yes, we can hide the latency of the floating point instructions by moving the add instructions in between floating point and store instructions – we’d only need 3 threads. Moving the third load up to follow the second load would further reduce thread requirement to only 2.
Problem M3.10: Multithreading [?? Hours]

Problem M3.10.A

Fixed Switching: ________6________ Thread(s)

If we have N threads and L.D. executes in cycle 1, FADD, which depends on the load executes in cycle $2N + 1$. To fully utilize the processor, we need to hide 12-cycle memory latency, $2N + 1 \geq 13$. The minimum number of thread needed is 6.

Data-dependent Switching: ________4________ Thread(s)

In steady state, each thread can execute 4 instructions (FADD, BNE, LD, ADDI). Therefore, to hide 11 cycles between ADDI and FADD, a processor needs $\lceil\frac{11}{4}\rceil + 1 = 4$ threads.

Problem M3.10.B

Fixed Switching: ________2________ Thread(s)

Each FADD depends on the previous iteration's FADD. If we have N threads and the first FADD executes in cycle 1, the second FADD executes in cycle $4N + 1$. To fully utilize the processor, we need to hide 5-cycle latency, $4N + 1 \geq 6$. The minimum number of thread needed is 2.

Data-dependent Switching: ________2________ Thread(s)

In steady state, each thread can execute 4 instructions (FADD, BNE, LD, ADDI). Therefore, to hide 2 cycles between ADDI and FADD, a processor needs $\lceil\frac{2}{4}\rceil + 1 = 2$ threads.
Problem M3.10.C

Consider a Simultaneous Multithreading (SMT) machine with limited hardware resources. Circle the following hardware constraints that can limit the total number of threads that the machine can support. For the item(s) that you circle, briefly describe the minimum requirement to support \( N \) threads.

(A) Number of Functional Unit  
Since not all the threads are executed in each cycle, the number of functional unit is not a constraint that limits the total number of threads that the machine can support.

(B) Number of Physical Registers  
We need at least \([N \times \text{(number of architecture registers)} + 1]\) physical registers.

(C) Data Cache Size  
This is for performance reasons.

(D) Data Cache Associatively  
This is for performance reasons.