Problem 1: Definitive Properties (do 6 of 10 for 36 pts)

Please explain what these things are or what the terms mean, for full credit. Partial credit will be given for examples and for partial answers. However, our intent is for you to capture a complete definition of (6) six of the following terms in the context of our our course. If you do more, we will only give credit for the first six (6) we see.

FOR FULL CREDIT, YOUR ANSWER MUST INCLUDE THE ITEM NUMBER, THE TERM THAT YOU ARE DEFINING, AND A COMPLETE DEFINITION, IN THE EXAM BOOKLET. A NON-REFUNDABLE FEE OF UP TO (4) POINTS WILL BE ASSESSED FOR FAILING TO FOLLOW THESE INSTRUCTIONS

1.1 clock rate
1.2 benchmark
1.3 register-memory architecture
1.4 resource contention
1.5 immediate operand
1.6 effective address
1.7 branch penalty
1.8 interstage register
1.9 instruction issue
1.10 precise exception handling

Problem 2: Duplicated Processors (do 3 of 6 for 18 pts)

The following questions are about Meep and Moop, two experimental machines that Muffin, the stuffed owl, is benchmarking when she’s not guarding her mistress, the lovely Jalenstrix. Some of the questions use the information in the table below; so, read carefully before assuming that Muffin left out some crucial pellet of information.

Do exactly three (3) of the six (6) problems below. Be sure to write down the number of the problem that you are doing in your exam booklet.

<table>
<thead>
<tr>
<th></th>
<th>Meep</th>
<th>Moop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floating Point IC</td>
<td>$2 \times 10^6$</td>
<td>$3 \times 10^6$</td>
</tr>
<tr>
<td>Integer IC</td>
<td>$198 \times 10^6$</td>
<td>$2 \times 10^6$</td>
</tr>
<tr>
<td>Average CPI</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Clock Cycle Time</td>
<td>$2 \times 10^{-9}$</td>
<td>$2 \times 10^{-12}$</td>
</tr>
</tbody>
</table>
2.1 Answer these three questions about the MEEP machine.

2.1.a Write an expression for the performance of MEEP.
2.1.b Write an expression for the MIPS rating of MEEP.
2.1.c Is the MIPS rating a reliable predictor of performance for a given machine? Explain your answer for full credit.

2.2 Suppose that you modify the MOOP so that the average CPI increases by 25% and the IC decreases by 30%. How much faster must the new clock chip be to increase the CPU performance rate? Explain your answer clearly for full credit.

2.3 Suppose that we give MEEP a new optimizing compiler. On the average, should we expect the IC to decrease? Explain your answer for full credit.

2.4 Suppose ALU instructions on MOOP take 2 clock cycles (cc), jumps take 1 cc, and all other instructions take 3 cc’s. Answer the following questions

2.4.a If 40% of the instructions are ALU instructions and 10% are jumps, what is the average CPI?
2.4.b Suppose that the MOOP optimizing compiler decreases ALU instructions to 20% and drops the number of jumps to 7%, without modifying the frequencies of any other instructions. What change, if any, would you expect in the execution time of a program on MOOP? Justify your answer for full credit.

2.5 An enhancement of MEEP gives a speedup of 10 whenever it can be used in a program. Suppose that the enhancement is used half of the time when the enhanced mode is in use. Answer these two questions regarding MEEP performance under this enhancement.

2.5.a What is the overall speedup from this enhancement?
2.5.b What percent of the original MEEP code was enhanced?

2.6 Suppose that we modify the MOOP ISA to decrease the CPI. Should we expect to see a decrease in the average execution time? Why or why not?
THE REMAINDER OF THE TEST DEALS WITH THE FOLLOWING MIPS ARCHITECTURE. SO DON’T SKIP THIS SECTION.

MMM-MIPS Specifications

The MMM-MIPS, Meesh’s Maddening Magnificent-MIPS architecture, has an unquenchable thirst for MIPS, the ISA of choice for CMSC 311 and CMSC 411. The MMM-MIPS is similar to the Multi-Cycle MIPS machine discussed in class, with the same five stages, and the EX stage consisting of four separate sub-pipes for integer and floating point ALU execution.

You must assume that the MMM-MIPS implementation includes the following features. Be sure to write down any additional assumptions you make on this exam paper.

- Separate instruction and data memories are used.
- Register reads and writes are split across a single clock cycle.
- Forwarding, bypassing, short circuiting, and load interlocks are implemented.
- Branches are resolved in the ID stage.
- Execution functional units satisfy Table FU (below).
- Execution functional units are fully pipelined wherever possible. That is, each pipeline stage takes one clock cycle.

<table>
<thead>
<tr>
<th>Unit</th>
<th># of Clock Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT ALU</td>
<td>1</td>
</tr>
<tr>
<td>FP ADD</td>
<td>8</td>
</tr>
<tr>
<td>FP MULT</td>
<td>18</td>
</tr>
<tr>
<td>FP DIV</td>
<td>60</td>
</tr>
</tbody>
</table>

FU: EX Stage Functional Unit Table

Problem 3: Delayed Principles (25 points)

3 Fill in the following table regarding the latencies and initiation intervals of each functional unit. Each correct value is worth one point.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Latency</th>
<th>Initiation Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT ALU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP ADD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP MULT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP DIV</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem 4: Disturbed Programming (33 points)

All problems refer to the the code fragment below, assuming that it is executed on the MMM-MIPS architecture defined previously. To save you from turning back and forth between pages, I’ll replicate the code fragment so you can see it every time you turn the page. And, don’t bother checking. They are identical.

The MIPS Code Fragment for Problem 4.

(1) DADDI R1, R1, #31000 ;
(2) LD R1, 88(R1)
(3) LD R2, 96(R1) ;
(4) BEQZ R2, done ;
(5) reload: LD F2, 0(R1) ;
(6) LD F4,8(R1) ;
(7) ADD.D F4, F4, F4 ;
(8) ADD.D F2, F2, F2 ;
(9) DIV.D F2, F4, F2 ;
(10) S.D F4, 8(R1) ;
(11) S.D F2, 0(R1) ;
(12) DADDI R1, R1,#-16 ;
(13) DADDI R2, R2,#-8 ;
(14) BNEZ R2, reload ;
(15) done: DADDI R8, R8, #12348 ;

Part One: Hazardous Analysis (15 pts)

For each of the specifications listed below, write down the number of an instruction (or pair of instructions and a register), that satisfies the request. If no such instruction is present, but can occur in MMM-MIPS, write NONE. If it is impossible for that request to be satisfied in MMM-MIPS, explain why not.

4.1 A control hazard.

4.2 A RAW hazard.

4.3 A potential WAW hazard.

4.4 A potential structural hazard.

4.5 A WAR hazard.
The MIPS Code Fragment for Problem 4.

(1) DADDI R1, R1, #-31000 ;
(2) LD R1, 88(R1)
(3) LD R2, 96(R1) ;
(4) BEQZ R2, done ;
(5) reload: LD F2, 0(R1) ;
(6) LD F4, 8(R1) ;
(7) ADD.D F4, F4, F4 ;
(8) ADD.D F2, F2, F2 ;
(9) DIV.D F2, F4, F2 ;
(10) S.D F4, 8(R1) ;
(11) S.D F2, 0(R1) ;
(12) DADDI R1, R1,#-16 ;
(13) DADDI R2, R2,#-8 ;
(14) BNEZ R2, reload ;
(15) done: DADDI R8, R8, #12348 ;

Part Two: Short Answers (18 points)

Briefly answer the following questions regarding the MIPS code fragment and the MMM-MIPS architecture. Be sure to explain your answers for full credit.

4.6 Are there any assumptions that must be made to assure that the code executes correctly? If so, give one. If not, then answer “NONE”.

4.7 Could the offset corresponding to the label done in instruction (4) be stored at address V in instruction memory, where V is 3 mod 4.

4.8 Could an exception occur while executing instruction (14)? If so, give an example of one. If not, explain why not.

4.9 Do you think that this code was generated by an optimizing compiler? Why or why not?

4.10 Would it be reasonable to modify the MMM-MIPS ISA to include two new branch types described below?

```
BLT   R1, R2, label ; Go to label if R1 < R2.
BLE   R1, R2, label ; Go to label if R1 ≤ R2.
```

4.11 Based on this code fragment, do you believe that the MMM-MIPS uses a branch delay slot? Why or why not?