<table>
<thead>
<tr>
<th>Problem</th>
<th>Score</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
<td></td>
<td>50 pts</td>
</tr>
<tr>
<td>Problem 2</td>
<td></td>
<td>30 pts</td>
</tr>
<tr>
<td>Problem 3</td>
<td></td>
<td>20 pts</td>
</tr>
</tbody>
</table>

Alias: ___________________________  Exam Total: _______ (100 pts)

*Good Luck!*
Problem 1: In the diagram below some wires are labeled with cycle numbers and corresponding values. For example, \( C2:4 \) indicates that at cycle 2 the pointed-to wire will hold a 4. Other wires are labeled just with cycle numbers, indicating that the wire is used at that cycle. If a value on any labeled wire is changed the code would execute incorrectly. There are no stalls during the execution of the code. The first instruction (\texttt{lui}) is shown (but don’t forget to finish it). In one of the \( C4 \)’s four bits are not shown, indicated by question marks. Note: \( C3:4 \) was not in the original exam; it conveys the same information as \( C2:4^* \). See solution. [50 pts]

- Write a program consistent with these labels.
- Fill in the blank next to \( C5: \).
- It is possible to find every register and immediate (though some are tricky). Use \texttt{r20-r29} for guessed registers.

Insn Addr | Cycle: 0 1 2 3 4 5 6 7 8
---|---
0x1000: lui | IF ID EX ME WB

---

---

---

---

Cycle: 0 1 2 3 4 5 6 7 8
Problem 2: The pipeline below includes a left shift unit (shifter) to be used by the left shift instructions. It performs the operation $x = v << a$ (left shift $v$ by $a$ bits), where $x$, $v$, and $a$ are the shifter ports shown in the diagram. [15 pts]

(a) Add connections to the shifter so that it can be used by shift left instructions. The code below should execute without a stall. Do not show control logic.

- The modifications should make efficient use of hardware.
- Be sure to show bit ranges on wires.
- Please check the code for dependencies.

```
# Cycle 0 1 2 3 4 5 6
sll  r1, r2, 10 IF ID EX ME WB
sllv r4, r5, r1 IF ID EX ME WB
sllv r6, r4, r1 IF ID EX ME WB
# Cycle 0 1 2 3 4 5 6
```

**Hint:** All of the shift instructions are Format R. The $rt$ register holds the value to be shifted. The $v$ in $sllv$ means the amount to shift is taken from a register.

<table>
<thead>
<tr>
<th>Opcode</th>
<th>rs</th>
<th>rt</th>
<th>rd</th>
<th>sa</th>
<th>func</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format-R:</td>
<td>31</td>
<td>26</td>
<td>21</td>
<td>16</td>
<td>15</td>
</tr>
</tbody>
</table>
Problem 2, continued: The EX stage is the right stage for the shifter. [15 pts]
(b) Suppose the shifter were placed in the ID stage and correctly connected, including bypasses.

What would be the impact on performance? Explain.

(c) Suppose the shifter were placed in the MEM stage and correctly connected, including bypasses.

What would be the impact on performance? Explain. 
*Hint: The explanation can use a code example.*
Problem 3: Answer each question below.

(a) Consider a new ISA, MIPS-DS2, in which branches and other control transfers have two delay slots, not just one. [10 pts]

Would the performance on something similar to the familiar (to us) five-stage implementation be higher, lower, or about the same as MIPS on its familiar five-stage implementation? Explain. (See next question before answering.)

Describe an implementation that might favor MIPS-DS2 and explain how it favors MIPS-DS2.

Other than the people that suggested the second delay slot and designed the hardware, if it works who gets the promotion, if it does not work who gets re-assigned to tech support (punished)? (Give a job description, not a name.) Hint: That person would probably quit if it were a MIPS-DS9. Note: The parenthesized part and the hint were not on the original exam. This question was originally the second of these three questions and seemed to refer to the five-stage implementation. Since the question really applied to the new implementation, not the five-stage one, the class was told not to answer it.
(b) For an instruction that raises a precise exception the handler can return and so the faulting instruction can be re-executed or skipped. [10 pts]

☐ Describe a situation in which the handler would need to either re-execute or skip. Mention what caused the exception and what the handler did about it.

☐ Other times there is no need to return. Describe such a situation, mention what caused the exception and why the handler would not return.