Q1. Architecture vs Microarchitecture

**True** or **false**: The following is architecturally visible (exposed by the architecture)?

1. Register file entries in a classical RISC pipeline
2. The stack in a stack architecture
3. Pipeline registers
4. Branch-delay / load-delay slots
5. NOPs
6. Pipeline bubbles
7. Condition codes, status flags
8. Memory address width
9. Instruction/data caches

Q2. Microcoded vs Pipelined

1. How does a microcoded machine differ from a classic RISC pipeline?

2. Why is a simpler microarchitecture generally possible with microcoding?
Q3. Microprogramming

Implement a conditional memory-to-memory move instruction in microcode for the single-bus RISC-V machine described in Handout #1. The instruction has the following format:

\[
\text{CMOV} \ (rd), \ (rs1), \ rs2
\]

CMOV performs the following operation: If the value in rs2 is true (non-zero), then the memory word loaded from the address in rs1 is stored to the address in rd.

\[
\text{if } R[rs2] \neq 0 \\
\quad M[rd] := M[rs1]
\]

Fill in the following table with the microinstructions and control signals. Optimize your microprogram to minimize the number of cycles and to set entries to don’t-cares (*) wherever possible.
<table>
<thead>
<tr>
<th>State</th>
<th>Pseudocode</th>
<th>IdIR</th>
<th>Reg Sel</th>
<th>Reg Wr</th>
<th>en Reg</th>
<th>IdA</th>
<th>IdB</th>
<th>ALUOp</th>
<th>en ALU</th>
<th>Id MA</th>
<th>Mem Wr</th>
<th>en Mem</th>
<th>Imm Sel</th>
<th>en Imm</th>
<th>µBr</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>FETCH0</td>
<td>MA := PC; A := PC</td>
<td>*</td>
<td>PC</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>N</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>IR := Mem</td>
<td>1</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>*</td>
<td>0</td>
<td>S</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>PC := A + 4</td>
<td>0</td>
<td>PC</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>INC_A_4</td>
<td>1</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>0</td>
<td>D</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP0</td>
<td>µBr to FETCH0</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>0</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>*</td>
<td>0</td>
<td>J</td>
<td>FETCH0</td>
</tr>
<tr>
<td>CMOVM0:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>