1. (8 points) The stages of our Y86 pipeline are: **Fetch (F)**, **Decode (D)**, **Execute (E)**, **Memory (M)**, and **Write Back (W)**. For each of the following, circle the letter indicating the stage at which this is most likely to occur. *If there are multiple possible answers, state the best answer.*

(a) **F D E M W:** Value read in from data memory location.

(b) **F D E M W:** Value written to data memory location.

(c) **F D E M W:** Effective address computed with displacement.

(d) **F D E M W:** Value read from register.

(e) **F D E M W:** Value written to register.

(f) **F D E M W:** Instruction read from instruction memory.

(g) **F D E M W:** Values added in ALU.

(h) **F D E M W:** Condition codes are computed.

2. (2 points) Assume that the individual stages of a datapath implementation have the latencies shown (including time to latch values):

<table>
<thead>
<tr>
<th>Fetch</th>
<th>Decode</th>
<th>Execute</th>
<th>Memory</th>
<th>Write-Back</th>
</tr>
</thead>
<tbody>
<tr>
<td>200ps</td>
<td>120ps</td>
<td>190ps</td>
<td>400ps</td>
<td>100ps</td>
</tr>
</tbody>
</table>

What is the shortest possible clock cycle time in a pipelined and non-pipelined implementation of this processor?

(a) _______  Pipelined  (b) _______  Non-pipelined

Page total: _____/10
3. (2 points) The pipelined Y86 tries to predict the next PC, *except for*:
   
   (a) calls
   (b) returns
   (c) unconditional jumps
   (d) conditional jumps
   (e) moves

4. (2 points) Pipelining a processor implementation probably *won’t* do which of the following:
   
   (a) increase throughput;
   (b) decrease latency;
   (c) allow a faster clock rate;
   (d) all of the above probably will happen.

5. Consider the following snippet of x86-64 code:

   400544: callq 400580
   40054D: movq %rax, (%rbx)

   with the following values in locations:

   Register %rsp 0x120
   Register %rip 0x400544

   What will be the values respectively of %rsp, %rip and the top of the stack following the next instruction execution?

   (a) 0x128, 0x400580, 0x40054D
   (b) 0x118, 0x400580, 0x40054D
   (c) 0x128, 0x40054D, 0x400580
   (d) 0x118, 0x40054D, 0x400580
   (e) none of these