Question 1

For each of the following x86-64 instructions, indicate whether the instruction could cause a page fault, whether it could cause a cache miss, and whether it could cause the dirty bit in the cache to be set to 1.

HINT: Think of whether a particular instruction could result in a read or write from main memory, the cache or the registers.

a. movq $7, %rcx

   Page fault? YES or NO | Cache miss? YES or NO | Dirty bit? YES or NO

b. movq $7, (%rdx)

   Page fault? YES or NO | Cache miss? YES or NO | Dirty bit? YES or NO

c. movq (%rax), %rbx

   Page fault? YES or NO | Cache miss? YES or NO | Dirty bit? YES or NO

d. addq %rbx, -8(%rax)

   Page fault? YES or NO | Cache miss? YES or NO | Dirty bit? YES or NO

Question 2

For the rest of the assignment, you will be tracing memory accesses in a system with the following architecture:

- 8-bit virtual addresses
- 16-byte page size
- 4 pages of physical RAM

a. How many bytes of data can a single process store in:

   (a) physical memory   (b) virtual memory
b. For each the given virtual addresses, divide the address into the **page number** and **page offset**.

\[
\begin{align*}
10001010 & \quad 10011111 & \quad 10101010
\end{align*}
\]

c. On the accompanying Page Table diagram, show the results of the following memory operations on RAM and the Page Table. Assume first-in-first-out replacement. You can also assume that there is only one process accessing memory during these 10 memory operations.

Within each box, time should progress downward, so the first address loaded appears at the top and subsequent changes are written below. To the right of the table, label each change with number of the operation that caused it. Annotate each operation below with *hit* or *page fault* to indicate whether the data was found in physical memory. Don’t forget to update the valid bits, especially when a page is kicked out!

\[
\begin{align*}
(a) \quad \text{read} & \quad 00011010 & \quad (f) \quad \text{write} & \quad 00001001 \\
(b) \quad \text{write} & \quad 00011011 & \quad (g) \quad \text{read} & \quad 00000000 \\
(c) \quad \text{read} & \quad 11111000 & \quad (h) \quad \text{read} & \quad 01010111 \\
(d) \quad \text{read} & \quad 11111010 & \quad (i) \quad \text{write} & \quad 00011010 \\
(e) \quad \text{read} & \quad 01101000 & \quad (j) \quad \text{read} & \quad 00011101
\end{align*}
\]

Use the Page Table (on the next page) and the map of RAM (below) to help keep track of virtual memory.

<table>
<thead>
<tr>
<th>frame #</th>
<th>page #</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>index</td>
<td>V</td>
</tr>
<tr>
<td>-------</td>
<td>---</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>