### Introduction to Computer Architecture

**Reference Pages**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>first</td>
<td>rdi</td>
</tr>
<tr>
<td>second</td>
<td>rsi</td>
</tr>
<tr>
<td>third</td>
<td>rdx</td>
</tr>
<tr>
<td>fourth</td>
<td>rcx</td>
</tr>
<tr>
<td>fifth</td>
<td>r8</td>
</tr>
<tr>
<td>sixth</td>
<td>r9</td>
</tr>
</tbody>
</table>

#### Argument Register Table

<table>
<thead>
<tr>
<th>Argument</th>
<th>Register</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>first</td>
<td>rdi</td>
<td>0</td>
</tr>
<tr>
<td>second</td>
<td>rsi</td>
<td>1</td>
</tr>
<tr>
<td>third</td>
<td>rdx</td>
<td>0</td>
</tr>
<tr>
<td>fourth</td>
<td>rcx</td>
<td>0</td>
</tr>
<tr>
<td>fifth</td>
<td>r8</td>
<td>0</td>
</tr>
<tr>
<td>sixth</td>
<td>r9</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Logic Gates

**AND gate**

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>x * y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**OR gate**

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>x + y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**NOT gate**

<table>
<thead>
<tr>
<th>x</th>
<th>x'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**NAND gate**

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>(x * y)'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**NOR gate**

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>(x + y)'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
### Arithmetic/Logic:

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Source</th>
<th>Destination</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>adds</td>
<td>$imm/reg</td>
<td>%reg/mem</td>
<td>add</td>
</tr>
<tr>
<td>adds</td>
<td>mem</td>
<td>%reg</td>
<td>add</td>
</tr>
<tr>
<td>ands</td>
<td>$imm/reg</td>
<td>%reg/mem</td>
<td>bit-wise and</td>
</tr>
<tr>
<td>ands</td>
<td>mem</td>
<td>%reg</td>
<td>bit-wise and</td>
</tr>
<tr>
<td>cmps</td>
<td>$imm/reg</td>
<td>%reg/mem</td>
<td>compare</td>
</tr>
<tr>
<td>cmps</td>
<td>mem</td>
<td>%reg</td>
<td>compare</td>
</tr>
<tr>
<td>decs</td>
<td>%reg/mem</td>
<td></td>
<td>decrement</td>
</tr>
<tr>
<td>divs</td>
<td>%reg/mem</td>
<td></td>
<td>unsigned divide</td>
</tr>
<tr>
<td>idivs</td>
<td>%reg/mem</td>
<td></td>
<td>signed divide</td>
</tr>
<tr>
<td>imuls</td>
<td>%reg/mem</td>
<td></td>
<td>signed multiply</td>
</tr>
<tr>
<td>incs</td>
<td>%reg/mem</td>
<td></td>
<td>increment</td>
</tr>
<tr>
<td>leaw</td>
<td>mem</td>
<td>%reg</td>
<td>load effective address</td>
</tr>
<tr>
<td>muls</td>
<td>%reg/mem</td>
<td></td>
<td>unsigned multiply</td>
</tr>
<tr>
<td>negs</td>
<td>%reg/mem</td>
<td></td>
<td>negate</td>
</tr>
<tr>
<td>ors</td>
<td>$imm/reg</td>
<td>%reg/mem</td>
<td>bit-wise inclusive or</td>
</tr>
<tr>
<td>ors</td>
<td>mem</td>
<td>%reg</td>
<td>bit-wise inclusive or</td>
</tr>
<tr>
<td>sal</td>
<td>$imm/%cl</td>
<td>%reg/mem</td>
<td>shift arithmetic left</td>
</tr>
<tr>
<td>sars</td>
<td>$imm/%cl</td>
<td>%reg/mem</td>
<td>shift arithmetic right</td>
</tr>
<tr>
<td>shls</td>
<td>$imm/%cl</td>
<td>%reg/mem</td>
<td>shift left</td>
</tr>
<tr>
<td>shrs</td>
<td>$imm/%cl</td>
<td>%reg/mem</td>
<td>shift right</td>
</tr>
<tr>
<td>subs</td>
<td>$imm/reg</td>
<td>%reg/mem</td>
<td>subtract</td>
</tr>
<tr>
<td>subs</td>
<td>mem</td>
<td>%reg</td>
<td>subtract</td>
</tr>
<tr>
<td>tests</td>
<td>$imm/reg</td>
<td>%reg/mem</td>
<td>test bits</td>
</tr>
<tr>
<td>tests</td>
<td>mem</td>
<td>%reg</td>
<td>test bits</td>
</tr>
<tr>
<td>xors</td>
<td>$imm/reg</td>
<td>%reg/mem</td>
<td>bit-wise exclusive or</td>
</tr>
<tr>
<td>xors</td>
<td>mem</td>
<td>%reg</td>
<td>bit-wise exclusive or</td>
</tr>
</tbody>
</table>

$s = b, w, l, q; w = l, q$

### Conditional Jumps:

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Location</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>ja</td>
<td>label</td>
<td>jump above (unsigned)</td>
</tr>
<tr>
<td>jae</td>
<td>label</td>
<td>jump above/equal (unsigned)</td>
</tr>
<tr>
<td>jb</td>
<td>label</td>
<td>jump below (unsigned)</td>
</tr>
<tr>
<td>jbe</td>
<td>label</td>
<td>jump below/equal (unsigned)</td>
</tr>
<tr>
<td>je</td>
<td>label</td>
<td>jump equal</td>
</tr>
<tr>
<td>jg</td>
<td>label</td>
<td>jump greater than (signed)</td>
</tr>
<tr>
<td>jge</td>
<td>label</td>
<td>jump greater than/equal (signed)</td>
</tr>
<tr>
<td>jl</td>
<td>label</td>
<td>jump less than (signed)</td>
</tr>
<tr>
<td>jle</td>
<td>label</td>
<td>jump less than/equal (signed)</td>
</tr>
<tr>
<td>jne</td>
<td>label</td>
<td>jump not equal</td>
</tr>
<tr>
<td>jno</td>
<td>label</td>
<td>jump no overflow</td>
</tr>
</tbody>
</table>