Evaluation of our ISA

- x295++ with base + displacement mode + immediate
- sample C code: \( z = \frac{7}{15} (x + y) \times \frac{7}{15} (x - y) \)
- Let's count the memory accesses:

**fetch**
- COPY $x, r7 \rightarrow memory address 10_{10}
- LOAD 0(r7), r1 \rightarrow 7
- LOAD \( \frac{y-x}{2} \) r7, r2 \rightarrow 8
- COPY r2, r3 \rightarrow 8
- ADD r1, r2 \rightarrow 15
- SUB r1, r3 \rightarrow -1
- MUL r2, r3 \rightarrow -15
- STORE r3, \( (z-x) \) (r7)

**execute**

Total:

Assumptions: x, y, z are memory addresses at which we find values for x, y & z

- x is 9 bit wide
Evaluation of our ISA

- x295++ with base + displacement mode
- sample C code: \( z = (x + y) \times (x - y) \)
- Let's count the memory accesses:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>fetch</th>
<th>execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY $x, r7</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>LOAD 0(r7), r1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>LOAD (y-x)(r7), r2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>COPY r2, r3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ADD r1, r2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SUB r1, r3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>MUL r2, r3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>STORE r3, (z-x)(r7)</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Total: 8 + 3 = 11

Assumptions: • x, y and z are memory addresses at which we find values for x, y and z  
• x is 9 bit wide