LAB 3
OVERVIEW

• You will be building a matrix sum accelerator

**n-by-n Matrix**

```
1 2 3
4 5 6
7 8 9
n=3
```

**Row Sum**

```
1 2 3
4 5 6
7 8 9
rowSum(0) = 6
```

**Column Sum**

```
1 2 3
4 5 6
7 8 9
colSum(1) = 15
```
LAB 3
ACCELERATOR INSTRUCTIONS

• Three instruction types:
  
  • SETUP: Define a matrix size and starting memory address
  
  • ROWSUM: Find the sum of the given row
  
  • COLSUM: Find the sum of the given column

// Pseudo-assembly example
SETUP N=3 ADDR=0x10000
ROWSUM 1
ROWSUM 1
COLSUM 2
COLSUM 1
COLSUM 0
LAB 3

NAIVE IMPLEMENTATION

• Each cycle, access the next matrix element

• n cycles for each sum
  • Slower for cache misses

• Can we do better?

ROWSUM(0) = 6
LAB 3
BETTER IMPLEMENTATION

• Avoid repeated memory accesses!

• For each row sum, store partial column sums

• Second set of operations is now very fast

\[
\begin{array}{ccc}
\text{ROWSUM 0} &=& 6 \\
\text{ROWSUM 1} \\
\text{ROWSUM 2} \\
\text{COLSUM 0} \\
\end{array}
\]

Partial COLSUMs:
\[
\begin{array}{ccc}
C1 & C2 & C3 \\
1 & 2 & 3 \\
\end{array}
\]
LAB 3
BETTER IMPLEMENTATION

• Avoid repeated memory accesses!
• For each row sum, store partial column sums
• Second set of operations is now very fast

ROWSUM 0 = 6
ROWSUM 1 = 15
ROWSUM 2
COLSUM 0

1 2 3
4 5 6
7 8 9

Partial COLSUMs:
C1 C2 C3
5 2 9
LAB 3

BETTER IMPLEMENTATION

• Avoid repeated memory accesses!

• For each row sum, store partial column sums

• Second set of operations is now very fast

<table>
<thead>
<tr>
<th>Partial COLSUMs:</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>15</td>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

| ROWSUM 0         |  6 |
| ROWSUM 1         | 15 |
| ROWSUM 2         | 24 |

<table>
<thead>
<tr>
<th>COLSUM 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3</td>
</tr>
<tr>
<td>4 5 6</td>
</tr>
<tr>
<td>7 8 9</td>
</tr>
</tbody>
</table>
LAB 3
BETTER IMPLEMENTATION

• Avoid repeated memory accesses!

• For each row sum, store partial column sums

• Second set of operations is now very fast

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Partial COLSUMs:

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>15</td>
<td>18</td>
</tr>
</tbody>
</table>

ROWSUM 0 = 6
ROWSUM 1 = 15
ROWSUM 2 = 24
COLSUM 0 = 12
LAB 3
INTEGRATION AND TEST

- Interface with Chisel instantiations of the RoCC interface
- No actual Rocket core or assembly instructions
- Magic memory

![Diagram of Chisel TestHarness and Accelerator with memory interfaces and control signals](image-url)