## Booth’s Algorithm for Binary Multiplication Example

Multiply 14 times -5 using 5-bit numbers (10-bit result).

- 14 in binary: 01110
- -14 in binary: 10010 (so we can add when we need to subtract the multiplicand)
- -5 in binary: 11011

Expected result: -70 in binary: 11101 11010

<table>
<thead>
<tr>
<th>Step</th>
<th>Multiplicand</th>
<th>Action</th>
<th>Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>upper 5-bits 0, lower 5-bits multiplier, 1 “Booth bit” initially 0</td>
</tr>
<tr>
<td>0</td>
<td>01110</td>
<td>Initialization</td>
<td>00000  11011  0</td>
</tr>
<tr>
<td>1</td>
<td>01110</td>
<td>10: Subtract Multiplicand</td>
<td>00000 + 10010 = 10010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shift Right Arithmetic</td>
<td>10010  11011  0</td>
</tr>
<tr>
<td>2</td>
<td>01110</td>
<td>11: No-op</td>
<td>11001  01101  1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shift Right Arithmetic</td>
<td>11100  10110  1</td>
</tr>
<tr>
<td>3</td>
<td>01110</td>
<td>01: Add Multiplicand</td>
<td>11100 + 01110 = 010110 (Carry ignored because adding a positive and negative number cannot overflow.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shift Right Arithmetic</td>
<td>01010  10110  1</td>
</tr>
<tr>
<td>4</td>
<td>01110</td>
<td>10: Subtract Multiplicand</td>
<td>00101 + 10010 = 10111</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shift Right Arithmetic</td>
<td>10111  01011  0</td>
</tr>
<tr>
<td>5</td>
<td>01110</td>
<td>11: No-op</td>
<td>11011  10101  1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shift Right Arithmetic</td>
<td>11101  11010  1</td>
</tr>
</tbody>
</table>