**AutoBench™ Version 1.1**

**Benchmark Name: Finite Impulse Response (FIR) Filter**

**Benchmark Description**
This EEMBC benchmark algorithm simulates an embedded automotive/industrial application where the CPU performs a Finite Impulse Response (FIR) filtering sample on 16-bit or 32-bit fixed-point values. High- and low-pass FIR filters simply process the input signal data.

**Optimization Rules**

<table>
<thead>
<tr>
<th>Category</th>
<th>Allowed</th>
<th>Disallowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSI C</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Intrinsics/Language Extensions</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Custom Libraries</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Assembly Language</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>HW Accelerators</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Algorithm Flowchart**

1. Start
2. Initialize and Get Test Data
3. Check Input values
   - No
   - Yes
   - Compute FIR Low Pass Filter
   - Compute FIR High Pass Filter
4. Check final?
   - No
   - Yes
   - Cleanup, Report Results
5. Stop