Modifications to LC3
Subroutines Operations

Recall
How are functions called and returned from?
• Call cannot be a plain jump (LC4 JMP)
• Needs to remember return address
• So that callee function can return to proper PC in caller

JSR (Jump Sub-Routine)

16-Word Aligned Absolute Addressing
PC = (PC & 0x8000) | (UIMM11 << 4)
• Instruction memory split between user and OS
  ➢ User: x0000-x7FFF, OS: x8000-xFFFF
• User and OS cannot call each other’s functions
• Note: Linux and WindowsNT do this too
PC = (PC & 0x8000) | (UIMM11 << 4)
• Can specify any 16-word aligned target in your half of memory
• The cost is instruction memory “fragmentation” (unusable slots)
• Note: real ISAs with fixed-length instructions do this too

...
**JMPR**

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

**Semantics**

$PC = \text{regs}[RS]$

**Assembly mnemonics**

- `JMPR R7` ; $PC = \text{regs}[R7]$
- `RET` ; `JMPR R7` pseudo-instruction

**Note:**

In LC3, RET is actually an instruction that has the same opcode as JMPR but the RS value is hard coded to 111 (7)