What do we know about the LC-3?

**Memory**
- Address space is $2^{16} = 0...32,767$
- Addressability is 16 bits
  \[ 1 \text{ word } = 16 \text{ bits} \]

**CPU**
- ALU word length is 16 bits
- Data operations performed by the ALU: ADD, AND, NOT
- How many 16-bit data registers does the CPU have? 8 (R0...R7)
- How many 1-bit registers does the CPU have? 3 (N,Z,P) condition codes
- The address of the next instruction to be executed by the CPU is stored in PC/IP
- After an Instruction is fetched from memory, it is stored in IR

**LC-3 Instructions - What are each of these instructions?**

- 0001 ADD
- 0101 AND
- 0010 LOAD
- 0000 BRANCH
- 1111 HALT

---

- R2 ← R0 + R1
- R6 ← R6 + #1
- Branch if N.Z.P R6 + 217
  \[ PC = x4028 \]
  \[ 0100 \ 0000 \ 0010 \ 1000 \ 0000 \ 0001 \ 101 \ 1001 \]
  \[ 0100 \ 0000 \ 1100 \ 0000 \]
  \[ x4181 \]
- R3 ← R3 AND #0 = R3 ← φ
A Program!

Friday, March 26, 2021  9:30 AM

Compute: \( R_3 \leftarrow A \times B \) where \( A = 5 \), \( B = 4 \)

ADD, AND, NOT + Load + Branch instruction

DO: \( 5 + 5 + 5 + 5 \)

Algorithm

Assume \( A \) is in \( M[x3007] \) and \( B \) is in \( M[x3008] \)

Register allocation

\( R_1 \leftarrow A \)

\( R_2 \leftarrow B \)

\( R_3 \) the accumulating sum

0: Start at \( x3000 \)

1. \( R_1 \leftarrow M[x3007] \) load \( A \)
2. \( R_2 \leftarrow M[x3008] \) load \( B \)
3. \( R_3 \leftarrow \phi \)

4. \( R_3 \leftarrow R_3 + R_1 \)

5. \( R_2 \leftarrow R_2 - 1 \)

6. \( \text{while } R_2 \neq 0 \)

7. \( \text{HALT} \)
11 The Program

Friday, March 26, 2021    9:30 AM

x3000: 1. $R_1 \leftarrow M[x3007] = M[PC+6]$

x3001: 2. $R_2 \leftarrow M[x3008] = M[PC+6]$

x3002: 3. $R_3 \leftarrow 0$

repeat

x3003: 4. $R_3 \leftarrow R_3 + R_1$

x3004: 5. $R_2 \leftarrow R_2 - 1$

while $R_2 = 0$

x3005: 6. Branch if Not 0 to x3003 = PC-3

x3006: 7. Stop

x3007: 8. $A = 5$

x3008: 9. $B = 4$

---

Copyright © McGraw-Hill Education. Permission required for reproduction or display.

<table>
<thead>
<tr>
<th>Address</th>
<th>15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>x3000</td>
<td>0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0</td>
</tr>
<tr>
<td>x3001</td>
<td>0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0</td>
</tr>
<tr>
<td>x3002</td>
<td>0 1 0 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>x3003</td>
<td>0 0 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>x3004</td>
<td>0 0 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>x3005</td>
<td>0 0 0 0 1 0 1 1 1 1 1 1 1 1 0 1 0 1</td>
</tr>
<tr>
<td>x3006</td>
<td>1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0</td>
</tr>
<tr>
<td>x3007</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0</td>
</tr>
<tr>
<td>x3008</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0</td>
</tr>
</tbody>
</table>