Lecture 8. TOY Instructions

- A program is a sequence of instructions
- An instruction is a 16-bit word, interpreted in one of many possible ways
- 3 instruction ‘formats,’ 16 different instructions

<table>
<thead>
<tr>
<th>Format 1</th>
<th>Format 2</th>
<th>Format 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 12</td>
<td>11 8</td>
<td>7 4</td>
</tr>
<tr>
<td>op</td>
<td>dst</td>
<td>reg_1</td>
</tr>
</tbody>
</table>

**Format 1**
- 0: halt
- 1: add
- 2: subtract
- 3: multiply

**Format 2**
- 9: load
- A: store

**Format 3**
- 4: system call
- 5: jump
- 6: jump if less
- 7: jump indirect
- 8: jump and link
- B: load immediate
Format 1 Instructions

- Format 1 instructions are *register-to-register* instructions
  
  Interpret *dst*, *reg*\(_1\)*, and *reg*\(_2\)* as register numbers
  
  Take operands from *reg*\(_1\)* and *reg*\(_2\)*, and put the result in *dst*
  
  Example: \(1{234}_{16}\) means \(R_2 \leftarrow R_3 + R_4\)
  
  Stores the sum of the contents of registers *R*\(_3\) and *R*\(_4\) into register *R*\(_2\)

\[
\begin{array}{cccccc}
15 & 12 & 11 & 8 & 7 & 4 & 3 & 0 \\
\hline
\text{op} & \text{dst} & \text{reg}_1 & \text{reg}_2
\end{array}
\]

- \(2116_{16}\): \(R_1 \leftarrow R_1 - R_6\)
- \(3267\): \(R_2 \leftarrow R_6 \times R_7\)
- \(C512\): \(R_5 \leftarrow R_1 \oplus R_2\) exclusive OR
- \(D645\): \(R_6 \leftarrow R_4 \& R_5\) logical AND
- \(E056\): \(R_0 \leftarrow R_5 >> R_6\) shift right
- \(E764\): \(R_7 \leftarrow R_6 << R_4\) shift left
- \(0000\): halt
Format 2 Instructions

15  12|11  8|7  4|3  0
   op  dst  reg  con4

• Format 2 instructions are *memory operation* instructions
  
  Interpret *dst* and *reg* as register numbers, *con4* as a 4-bit *unsigned* constant
  
  Compute the *effective address* *reg + con4*

• *Load* copies a word *from memory* at the effective address *to register dst*

  2123_{16} means R₁ ← M[R₂ + 3]

  Copy the contents of the memory location specified by adding 3 to the contents of register R₂ to register R₁

• *Store* copies a word *from register dst to memory* at the effective address

  5765_{16} means M[R₆ + 5] ← R₇

  Copy the contents of register R₇ to the memory location specified by adding 5 to the contents of register R₆

• When *con4* is 0, load/store are sometimes called *indirect* load/store
## Format 3 Instructions

<table>
<thead>
<tr>
<th>15</th>
<th>12</th>
<th>11</th>
<th>8</th>
<th>7</th>
<th>4</th>
<th>3</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textbf{op}</td>
<td>\textbf{dst}</td>
<td>\textbf{con8}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Most of the format 3 instructions are \textit{control} instructions
  - Interpret \textit{dst} as a register number, \textit{con8} as an 8-bit \textit{unsigned} constant or address
  - Compute a \textit{jump address} as either \textit{con8} or \textit{dst}
  - Set PC to that address
  - Oddballs: system call (4) and load immediate (B)

- \textit{Load immediate} copies \textit{con8} to register \textit{dst}
  - $\text{B234}_{16}$ means $R_2 \leftarrow 34_{16}$ \quad set register R2 to \textit{34}_{16}
  - Use load immediate to copy the contents of a register to another register:
    - $\text{B000}_{16}$ means $R_0 \leftarrow 0$ \quad set R0 to 0
    - $\text{1320}_{16}$ means $R_3 \leftarrow R_2 + R_0$ \quad set R3 to $R_2 + R_0$

- \textit{System call} invokes actions that need special permission, like I/O
  - \textit{con8} specifies the system call ‘action code’, \textit{dst} may specify an operand
  - $\text{4402}_{16}$ writes the contents of R4 to the standard output
Jump Instructions

• **Jump** instructions change the PC to *con8*, or to the contents of *dst*

  jump

  \[ \text{PC} \leftarrow 62_{16} \]

  The next instruction will be taken from \( M[62_{16}] \)

  jump if less

  \[ \text{PC} \leftarrow 62_{16} \text{ if the contents of } R_3 < 0 \]

  jump indirect

  \[ \text{PC} \leftarrow R_5 \]

  The next instruction will be taken from the address in \( R_5 \)

  jump and link

  \[ R_4 \leftarrow \text{PC}, \text{PC} \leftarrow 62_{16} \]

  The contents of the PC (3B\(_{16}\)) are saved in \( R_4 \), then the PC is set to 62\(_{16}\)

  The next instruction will be taken from \( M[62_{16}] \)

  Used for *function linkage* — calls and returns

• All instructions of format 3 use a constant as one operand and a register or the program counter as the other operand.
Example: Bit Twiddling

• Set b_0 of R_4 to b_{10} \land b_3 from R_1, clear b_{1}–b_{15} in R_4

\[
\begin{align*}
R_4 &= ((R_1\ll 10) \land (R_1\ll 3)) \land 1; \\
1010 & 0111 0111 0010 \quad R_1 \\
0000 & 0000 0010 1001 \quad R_1\ll 10 \\
0001 & 0100 1110 1110 \quad R_1\ll 3 \\
0001 & 0100 1100 0111 \quad (R_1\ll 10) \land (R_1\ll 3) \\
0000 & 0000 0000 0001 \quad ((R_1\ll 10) \land (R_1\ll 3)) \land 1
\end{align*}
\]

Assuming R_1 is initialized to A772_{16}

\[
\begin{align*}
00: & \quad B000 \quad R_0 \leftarrow 00 \\
01: & \quad 1210 \quad R_2 \leftarrow R_1 + R_0 = A772 \\
02: & \quad 1310 \quad R_3 \leftarrow R_1 + R_0 = A772 \\
03: & \quad B50A \quad R_5 \leftarrow 0A \\
04: & \quad B603 \quad R_6 \leftarrow 03 \\
05: & \quad E225 \quad R_2 \leftarrow R_2 \gg R_5 = 0029 \\
06: & \quad E336 \quad R_3 \leftarrow R_3 \gg R_6 = 14EE \\
07: & \quad C323 \quad R_3 \leftarrow R_2 \land R_3 = 14C7 \\
08: & \quad B401 \quad R_4 \leftarrow 01 \\
09: & \quad D443 \quad R_4 \leftarrow R_4 \land R_3 = 0001
\end{align*}
\]
Example: Polynomial Evaluation

- Evaluate $ax^2 + bx + c = 2x^2 + 3x + 9$ at $x = 10$ ($239_{10} = EF_{16}$)

  Store the ‘data’ in locations 30–33_{16}

<table>
<thead>
<tr>
<th>Location</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>000A</td>
</tr>
<tr>
<td>31</td>
<td>0002</td>
</tr>
<tr>
<td>32</td>
<td>0003</td>
</tr>
<tr>
<td>33</td>
<td>0009</td>
</tr>
</tbody>
</table>

- Use Horner’s method: rewrite $ax^2 + bx + c$ as $(ax + b)x + c$

<table>
<thead>
<tr>
<th>Location</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>B330</td>
</tr>
<tr>
<td>11</td>
<td>9430</td>
</tr>
<tr>
<td>12</td>
<td>9531</td>
</tr>
<tr>
<td>13</td>
<td>3554</td>
</tr>
<tr>
<td>14</td>
<td>9632</td>
</tr>
<tr>
<td>15</td>
<td>1556</td>
</tr>
<tr>
<td>16</td>
<td>3554</td>
</tr>
<tr>
<td>17</td>
<td>9633</td>
</tr>
<tr>
<td>18</td>
<td>1556</td>
</tr>
<tr>
<td>19</td>
<td>4502</td>
</tr>
</tbody>
</table>

  - System call 2: print R5 = 00EF
  - HALT

- Polynomial evaluation for arbitrary $x$

  many applications, one *raison d’etre* for early computers