Topic for today:
A first look at the control unit

Processor Control Unit

- Role of processor control unit
  - Keeps operations synchronized
  - Make sure that bits flow to the correct components at the correct time
- How can we build this control unit?
  - Hardwired control
  - Microprogrammed control
- The result is the same – control signals!

Decoding opcodes

Recall that a single operation in our ISA typically involves a sequence of several register transfers. A single register transfer can be executed by setting various control lines either high or low.

*Early in the history of computers, each instruction involved just a single register transfer, and the opcode indicated the high/low settings.*

Questions:

How do you get the settings needed for a register transfer from a single opcode?

How can you get a sequence of transfers from a single opcode?
Processor Control Unit

• Remember the register transfer language description for each MARIE instruction?
  – Table 4.7
  – This is what the control unit manages.
• Each microoperation consists of a distinctive signal pattern that is interpreted by the control unit and results in the execution of an instruction
  – RTL for the ADD instruction
    \[
    \begin{align*}
    \text{MAR} & \leftarrow X \\
    \text{MBR} & \leftarrow M[\text{MAR}] \\
    \text{AC} & \leftarrow \text{AC} + \text{MBR}
    \end{align*}
    \]

• Each of MARIE’s registers and main memory have a unique address along the datapath.
• The addresses take the form of signals issued by the control unit.

How many signal lines does MARIE’s control unit need?

Memory Buffer Register (MBR) Closeup

• Two sets of three signals each.
• \{P_2, P_1, P_0\}
  – controls reading from memory or a register
• \{P_5, P_4, P_3\}
  – controls writing to memory or a register.
Processor Control Unit

- How does the control unit perform operations in sequence?
- Longest instruction is IND (look at RTL in Table 4.7)
- 7 steps
- Need a 3-bit counter wired to a 3-8 decoder
- Counter reset for shorter instructions
- Output of decoder is "timing" signals: $T_8 - T_7$

The entire set of MARIE’s control signals consists of:
- Register controls
  - $P_x$ through $P_7$
- ALU controls
  - $A_i$ through $A_0$
- Timing
  - $T_8$ through $T_7$
- Counter reset $C_0$

ADD Instruction Control

- ADD instruction RTL
  - $MAR \leftarrow X$
  - $MAR \leftarrow [MAR]$
  - $AC \leftarrow AC + MBR$
- After the add instruction is fetched, the address ($X$) is in the rightmost 12 bits of the IR
- IR datapath address is 7
- Raise signals $P_2$, $P_1$, and $P_0$ to read from IR
- $X$ is copied to the MAR
- MAR datapath address is 1
- Raise signal $P_6$ to write to MAR

<table>
<thead>
<tr>
<th>ALU Control Signals</th>
<th>$A_1$</th>
<th>$A_0$</th>
<th>ALU Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Do Nothing</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>AC ← AC + MBR</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>AC ← AC - MBR</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>AC ← 0 (Clear)</td>
</tr>
</tbody>
</table>
ADD Instruction Control

- Complete signal sequence for ADD instruction
  - P3 P2 P1 P0 T0: MAR ← X
  - P4 P3 T1: MBR ← M[MAR]
  - A0 P5 P1 P0 T2: AC ← AC + MBR
  - Cr T3: [Reset counter]

- These signals are ANDed with combinational logic to bring about the desired machine behavior.

Process Control Unit

- This signal pattern needs to be produced regardless of whether the processor uses hardwired or microprogrammed control

**Hardwired control**

In a hardwired approach, the opcode is decoded by a large set of digital circuits (decoders, multiplexers, etc.) to send the correct signals to the registers, ALU, and internal CPU bus controller. (This is all just Boolean algebra!)

Also, the clock signal is connected to a counting circuit to ensure that actions happen in the correct sequence.
This is the hardwired logic for MARIE's Add = 0011 instruction.

Note

In a hardwired approach, any change in the ISA requires a major redesign of the CPU circuitry.
Consider the following

Discuss

Suppose you are designing a hardwired control unit for a very small computerized device. This system is so revolutionary that the system designers have devised an entirely new ISA for it. Because everything is so new, you are contemplating including one or two extra flip-flops and signal outputs in the cycle counter. Why would you want to do this? Why would you not want to do this? Discuss the tradeoffs.