CONTROL UNIT DESIGN

Introduction

* To design a control unit we need to design an FSM
* When you design an FSM, this is what you need to specify:

\[
\begin{align*}
\text{Inputs} & \xrightarrow{\text{Next-state logic}} \text{FSM state} \xrightarrow{\text{Output logic}} \text{Outputs} \\
\text{Inputs} & \xrightarrow{\text{Next-state logic}} \text{FSM state} \xrightarrow{\text{Output logic}} \text{Outputs}
\end{align*}
\]

* We will see two approaches to design the combinational logic (next-state and output)
* An FSM can be represented in two ways:

1) State transition table

<table>
<thead>
<tr>
<th>Current state</th>
<th>Inputs</th>
<th>Next state</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2) State transition diagram:

\[\text{MAR} \leftarrow \text{PC}\]
\[\text{PC} \leftarrow \text{PC} + 1\]
\[\text{MDR} \leftarrow M[\text{MAR}]\quad R = 0\]
\[R = 1\]
\[\vdots\]
Hardwired control unit

* Using the state transition table, we can design the combinational logic using the techniques learnt so far
* For the state representation we can use a counter and the IR

![Diagram of control unit concepts]

Advantage: it can work at high speed
Disadvantage: rigid design
Microprogrammed control unit

Key insight: rewrite FSM diagram as flowchart

FSM diagram

\[
\begin{align*}
\text{MAR} & \leftarrow \text{PC} \\
\text{PC} & \leftarrow \text{PC} + 1 \\
\text{MDR} & \leftarrow \text{M}[\text{MAR}] \\
R & = 0 \\
R & = 1
\end{align*}
\]

Flowchart

\[
\begin{align*}
\text{MAR} & \leftarrow \text{PC} \\
\text{PC} & \leftarrow \text{PC} + 1 \\
\text{MDR} & \leftarrow \text{M}[\text{MAR}] \\
R & = 0 \\
R & = 1
\end{align*}
\]
FSM becomes microprogram that executes microinstructions.

LC-3 state machine was designed with microprogrammed control approach in mind.
First approach: use a Read-Only Memory (ROM)

Question: how do we handle conditional constructs? (E.g., wait for memory read, branch instruction)
Question: what about the DECODE phase, where the opcode indicates the next state?
Advantage: flexible, easy to alter instruction set
Disadvantage: slow compared to hardwired design