Lesson 14: Microprogrammed Control
Objective

- Understand the design of microprogrammed control unit to generate all sequences of the control signals using a control memory
- Learn how the control memory stores at successive addresses the data for generating the control signal values 0 or 1 stored at each address and generates the sequences by address changes from the start address to end address
- Learn how the start address for generating the control signals depend on the instruction at ID
Microprogram concept
Microprogram concept

- Hardware, with a large number of logic gates, does not directly execute an instruction.
- Instead, the hardware executes very simple set of micro-operations called microinstructions.
- Each instruction specifies a sequence of micro-operations that are used to implement the instruction.
Microprogram concept

- Each instruction translated into a short program of microinstructions by the hardware; similar to the way a compiler translates each instruction in a high-level language program into a sequence of assembly-language instructions.
Microprogrammed control execution

- Microprogrammed control memory generates a set of outputs for a given set of microinstructions (set of control-Signal sequences) stored in that
Using microprogramming, architects could build simple hardware and then microprogram that hardware to execute complex instructions.

- Requirement of limited the amount of hardware that could be built into the processor.
- Enables design of instruction sets with complex instructions at the reduced total number of microinstructions required to implement a program.
Microprogrammed Control Unit organization
Organization of control store memory

- $m$-distinct instructions from $I_0$ to $I_{m-1}$, there are $m$-distinct microprograms stored in memory $mp_0$ to $mp_{m-1}$, each having a distinct start address.
- To execute an instruction, a microprogrammed processor accesses this memory to locate the set of microinstructions required to implement the instruction, $I_i$ at the ID and then execute the microinstructions $mp_i$ for $I_i$ in a sequence of addresses from the start address.
An $m$-instruction processor having $m$-sets of inputs and $m$-microprograms stored at control ROM distinct start addresses.
Use of Control memory in place of hardwired control

- Hardware, with a large number of logic gates, does not directly execute an instruction.
- Instead, the hardware executes very simple micro-operations.
- Each instruction specifies a sequence of micro-operations that are used to implement the instruction.
- Each instruction translated into a short program of microinstructions by the hardware; similar to the way a compiler translates each instruction in a high-level language program into a sequence of assembly-language instructions.
Use of Control memory in place of hardwired control

- Hardware, with a large number of logic gates, does not directly execute an instruction.
- Instead, the hardware executes very simple micro-operations.
- Each instruction specifies a sequence of micro-operations that are used to implement the instruction.
- Each instruction translated into a short program of microinstructions by the hardware; similar to the way a compiler translates each instruction in a high-level language program into a sequence of assembly-language instructions.
Summary
We Learnt

- Micro-programmed control using control memory
- Starting address of a microprogram as per ID output
- m-distinct instructions from $I_0$ to $I_{m-1}$, there are m-distinct microprograms stored in memory $mp_0$ to $mp_{m-1}$, each having a distinct start address
We Learnt

- Steps implement by outputs of control ROM from a microprogram start address to end address
- Requires simpler hardware to implement a complex instruction set
End of Lesson 14 on Microprogrammed Control