



# **DIGITAL DESIGN & COMPUTER ORGANISATION**

## **Microprocessor Control Logic – 2**

**Maresh H.B.**

**Department of Computer Science & Engineering**

## Load and Jump Instructions

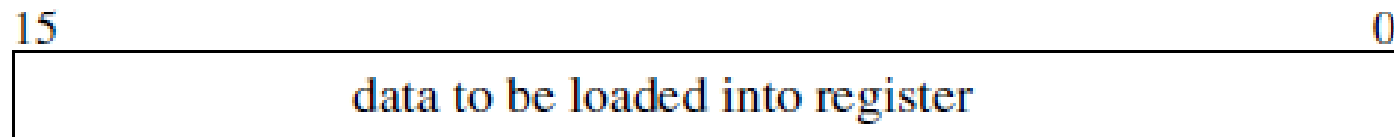
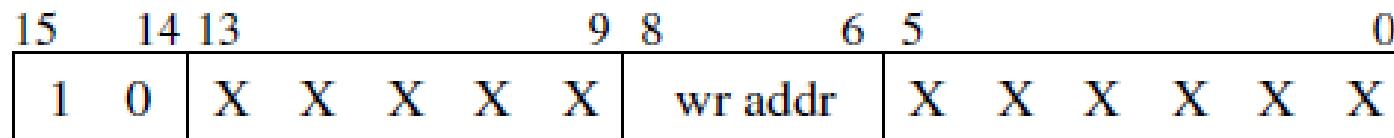
- ❖ The intent of this assignment is to enhance the control logic to implement a load and a jump instruction.
- ❖ The data-path has been augmented as follows.
- ❖ For the load instruction, data out from RAM has been connected to register file input.
- ❖ In order to enable above input to register file (instead of usual input from ALU) when a load instruction is being executed, the sel input of the register file is used, which is supplied by the control logic.

## Load and Jump Instructions

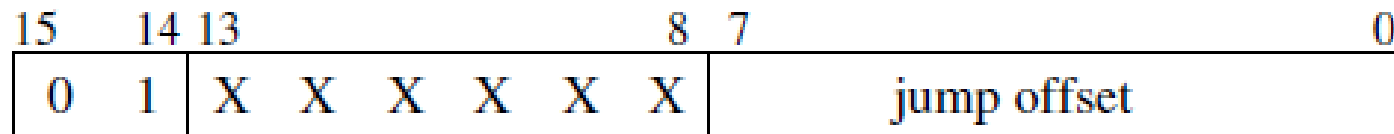
- ❖ For the jump instruction, bits in the instruction (IR) that represent the jump offset are connected to the offset input of the PC module, whose sub input is derived from the jump output supplied by the control logic.
- ❖ Only the sub input is used (and not the add input) since the jump is backwards only (offset is subtracted from PC).
- ❖ Also, the jump occurs only if the cout output of the ALU is high, ensured by AND of cout with jump in mproc module to generate sub.

## Load and Jump Instructions

❖ The format of the instructions is shown in figure.



Load instruction



Jump instruction

Figure: Format of load and jump instructions.

## Load and Jump Instructions

- ❖ The load instruction occupies two 16-bit words, the first of which specifies the address of the register (wr\_addr) into which the data is to be loaded, while the second contains the 16-bit data to be loaded.
- ❖ The jump instruction specifies an 8-bit offset.
- ❖ If cout is 1, then the offset is subtracted from the PC, else the next instruction is executed.

## Load and Jump Instructions

- ❖ Note that while ALU and jump instructions execute in two clock cycles, the load instruction (which loads a 16-bit word from memory) would require three clock cycles to execute.
- ❖ As a result, the FSM in the control logic would have three states.

## Load and Jump Instructions

- ❖ Only contents of control\_logic module need to be modified in this assignment.
- ❖ As mentioned above, the control logic now needs to supply two more outputs (compared to assignment 5 control logic) ***sel*** and ***jump***, which are to be asserted (for one or more clock cycles) during the load and jump instructions respectively.

## Load and Jump Instructions

- ❖ Also note that the logic used to generate pc\_inc inside the control logic needs to change.
- ❖ Previously, pc\_inc was asserted for only one of the three clock cycles required to execute an instruction, but for the load instruction pc\_inc needs to be asserted twice in three clock cycles.



## Load and Jump Instructions

- ❖ Note that even while executing a jump instruction, PC increment should be allowed to happen followed by offset subtract from PC (if cout is 1).
- ❖ Also, the inc and sub inputs to PC cannot be high at the same time, so the jump output of control logic (used to generate sub input to PC) must be accordingly generated.

## Microprocessor Control Logic – 2

## Load and Jump Instructions

❖ Sketch of control logic – 2.

