Royal Holloway University of London

Lecture 7 – Datapath Review

Karl R. Wilcox
Karl@cs.rhul.ac.uk

Royal Holloway University of London

Objectives

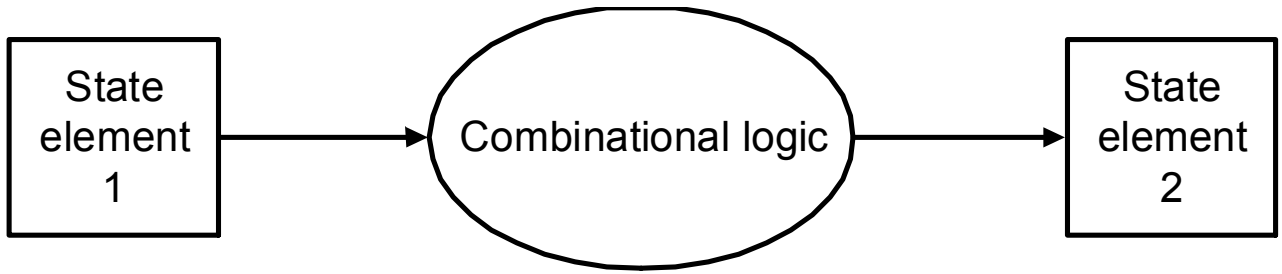
- **In this lecture we will cover**
  - **Review of a simple processor datapath**
    - **From Computer Engineering I**
  - **Reminder about extending this to a multi-cycle datapath**
  - **(diagrams are from Patterson & Hennessy)**

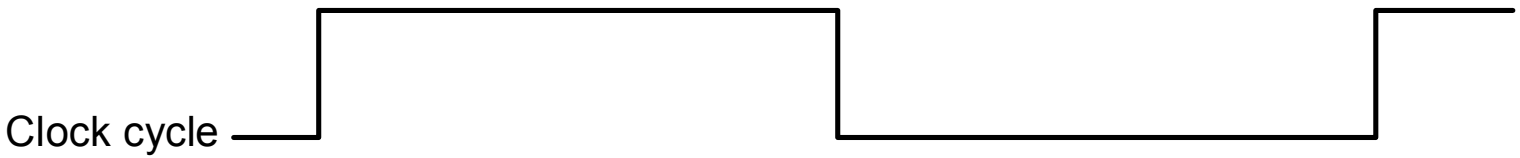
Datapath Purpose

- We have defined various instructions that we would like our processor to support
- To implement these instructions we require:
  - Various hardware components
    - Registers
    - Memory elements
    - Adders etc.
  - Data paths to connect these elements
  - Signals to control the routing of data and the operation of the hardware components

Clocking Methodology

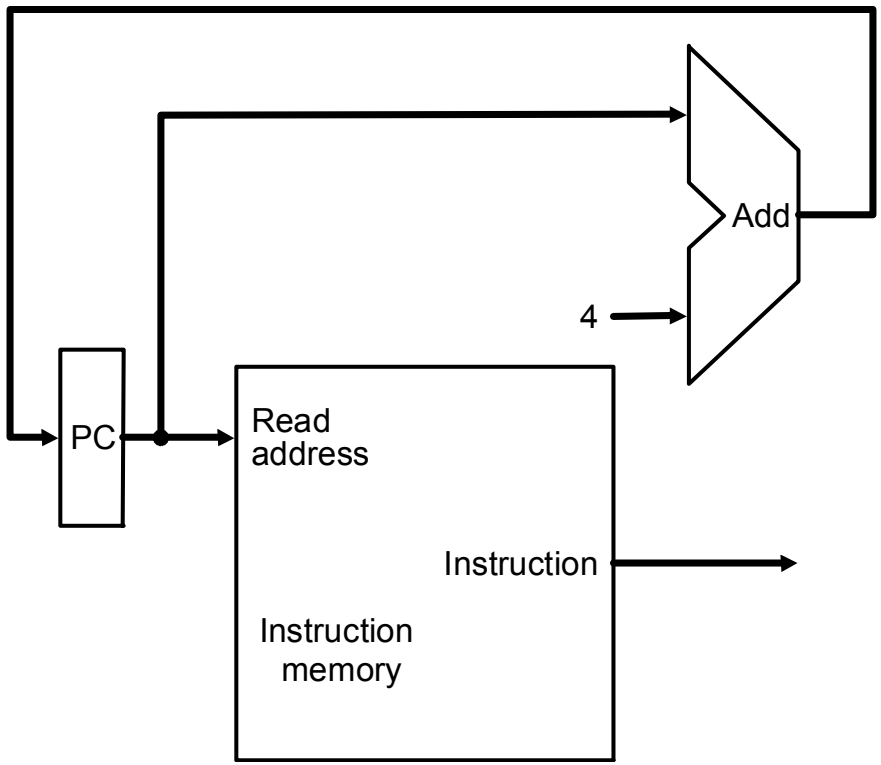
- **For our simple datapath we will assume that state elements are written on a positive clock edge**





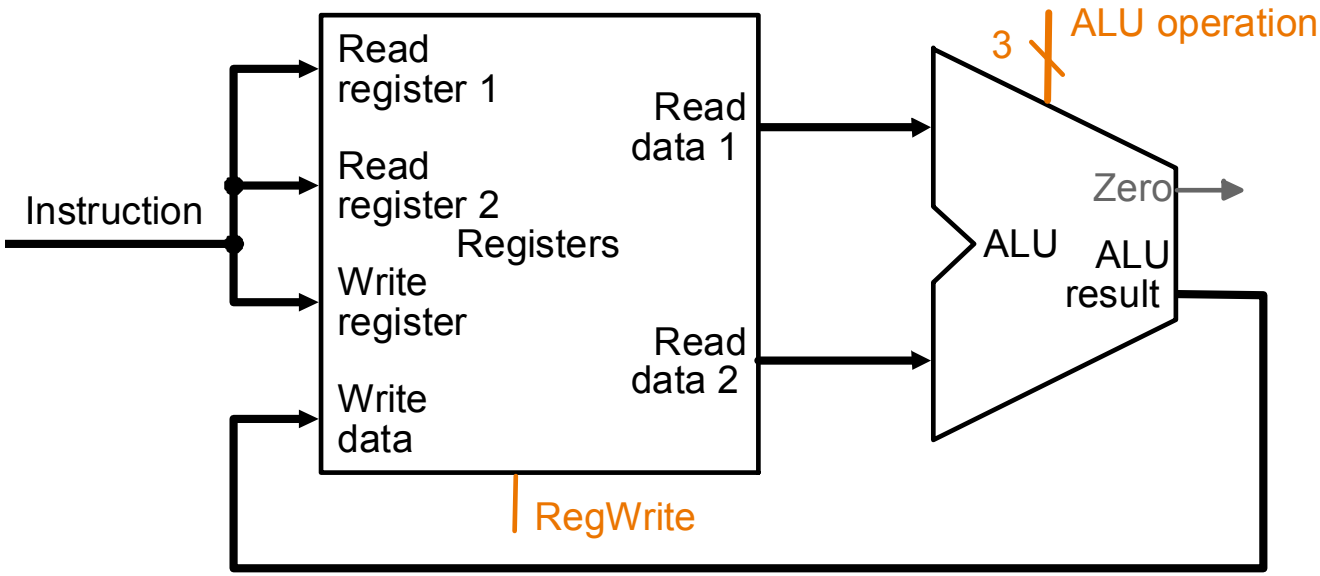
Fetching Instructions

- **We need a means of fetching instructions from memory and incrementing the program counter**



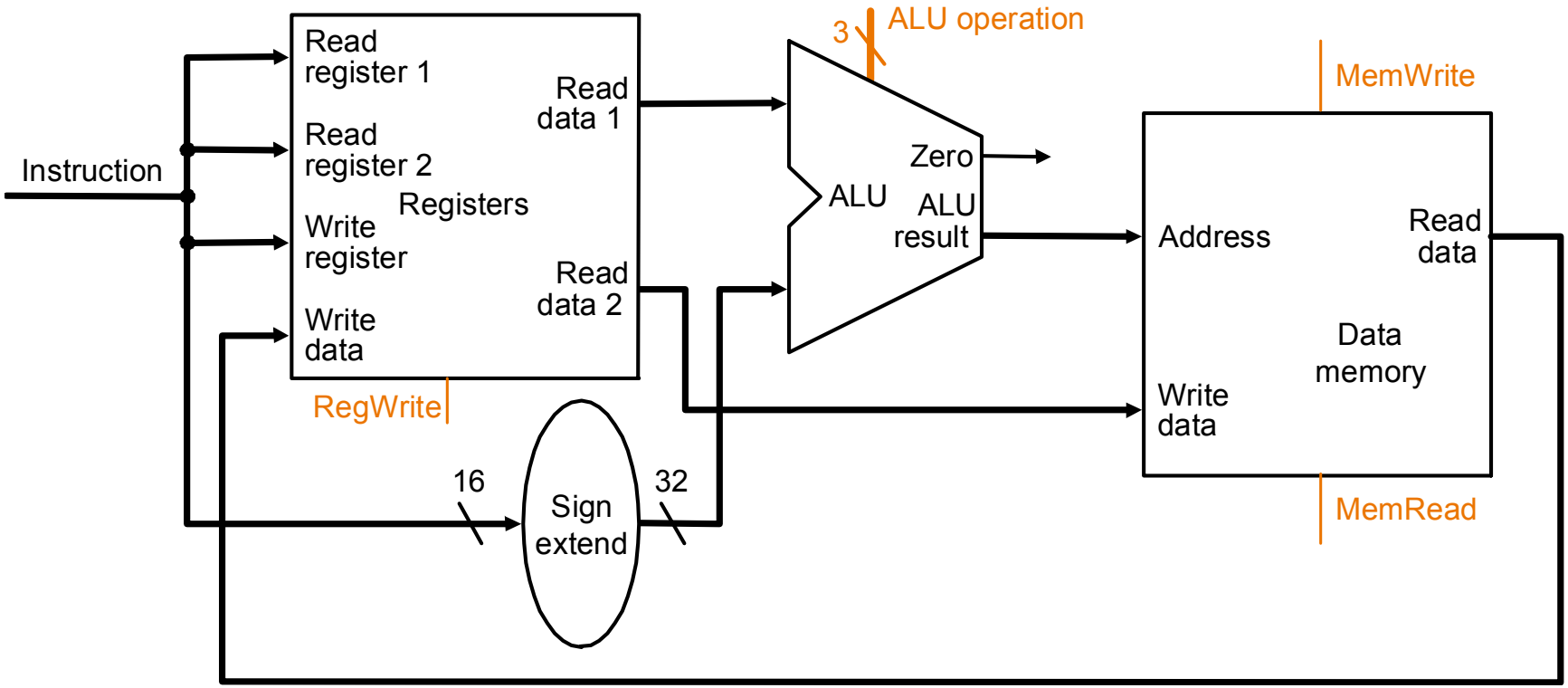
Register / Register instructions

- **We need a means to implement instructions such as**
  - **`add $t1, $t2, $t3`**
  - **(add registers 2 & 3, store result in register 1)**



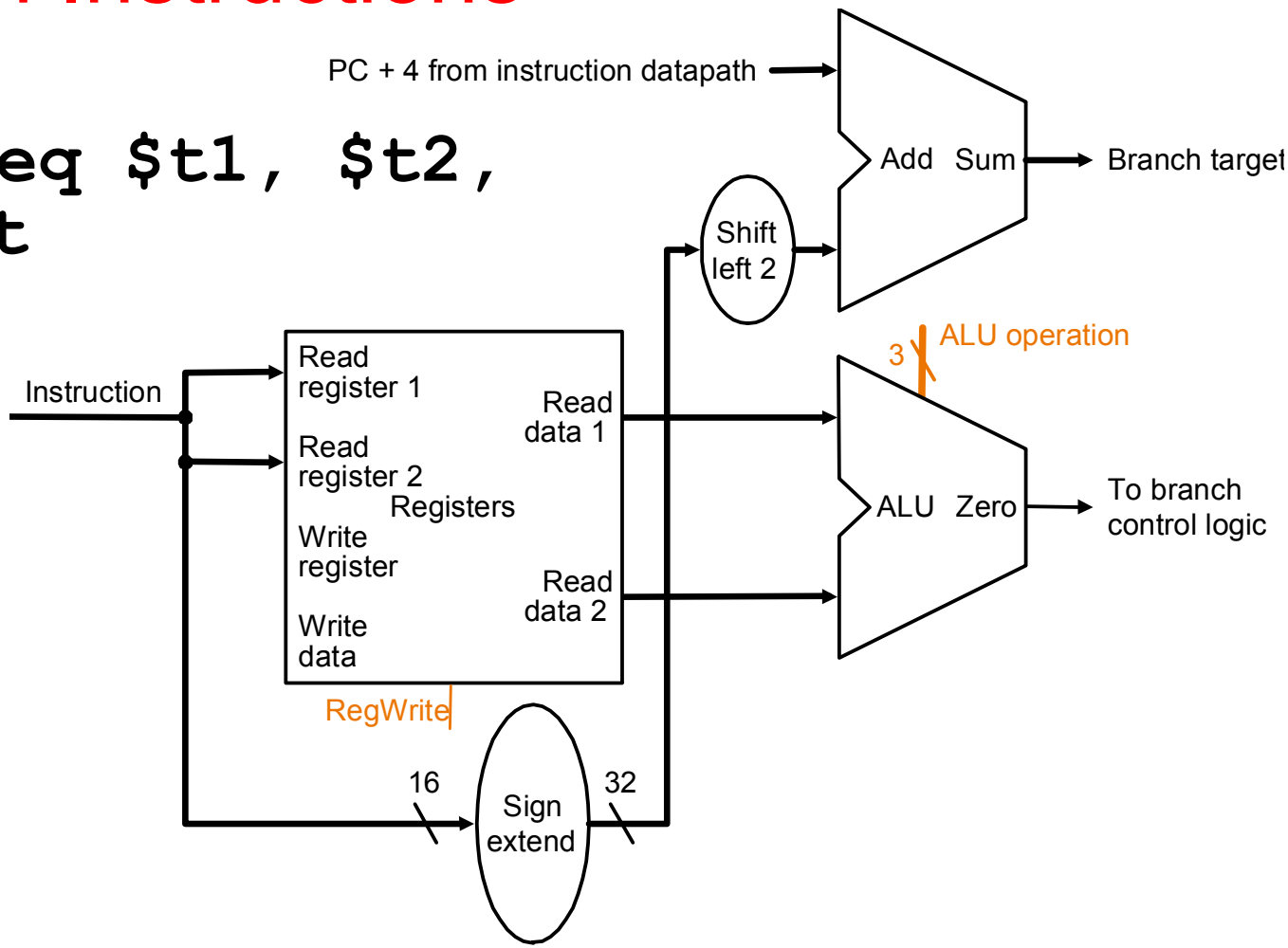
Load and Store Instructions

- **E.g. `lw $t1, offset($t2)`**



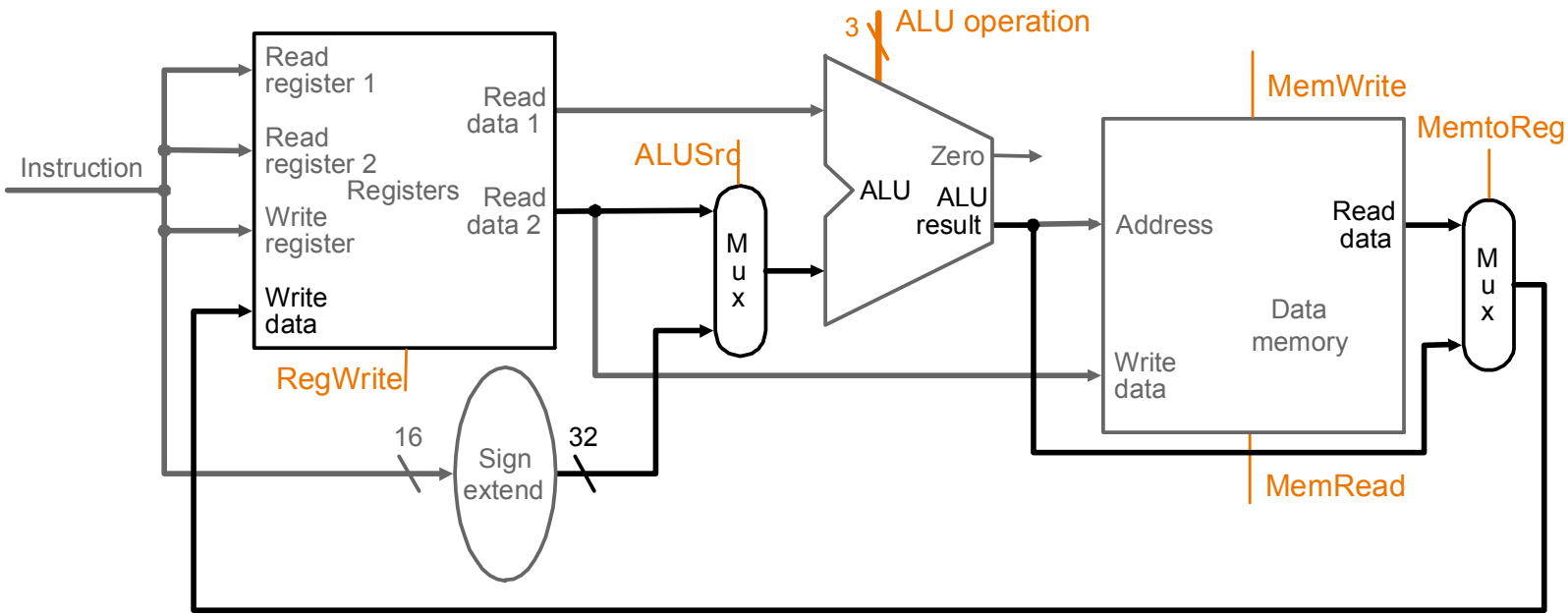
Branch Instructions

- **E.g. `beq $t1, $t2, offset`**

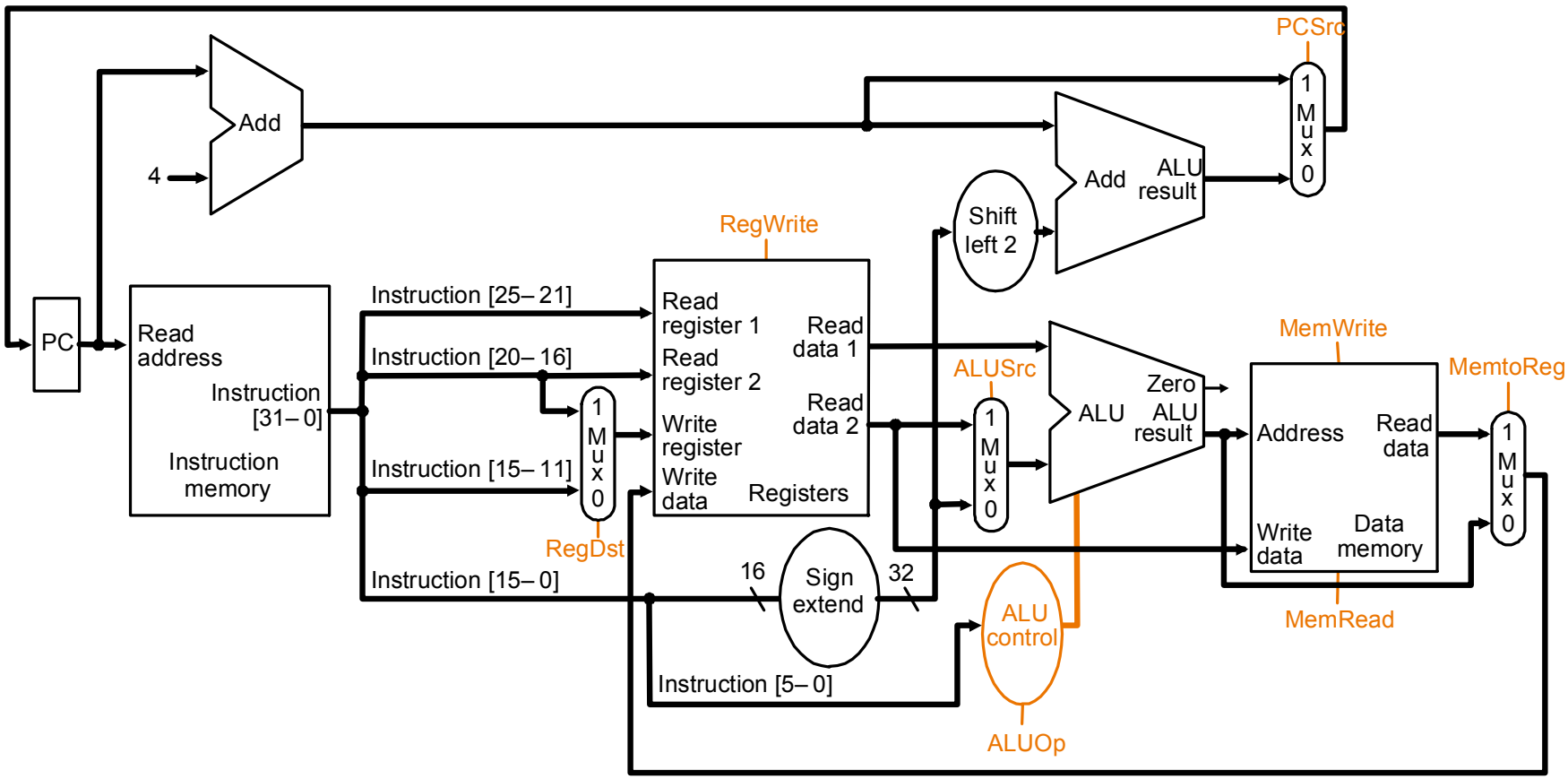


Combining Implementations

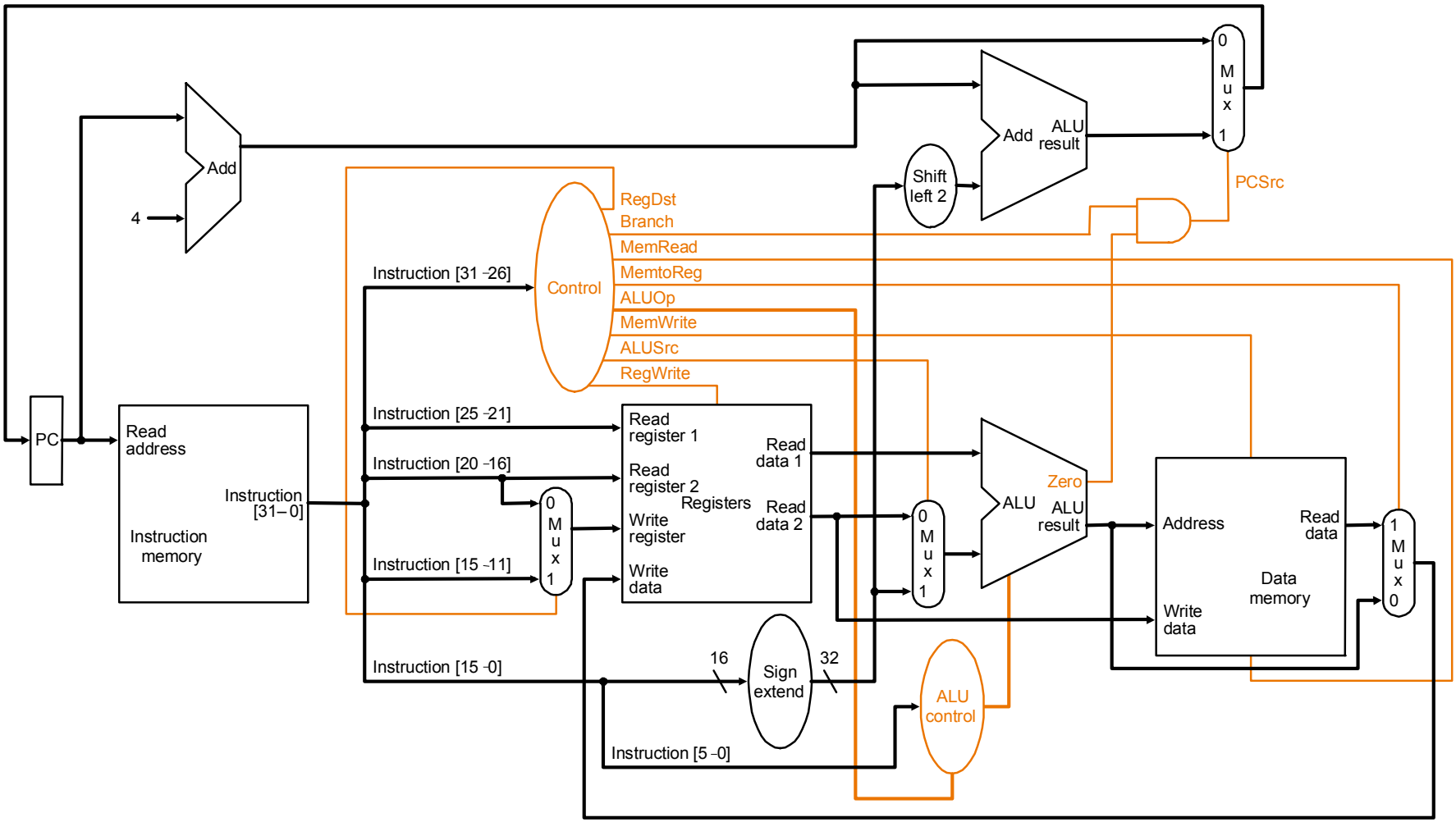
- **For example, combining memory access and register / register instructions**



Combining All Implementations



Generating Control Signals



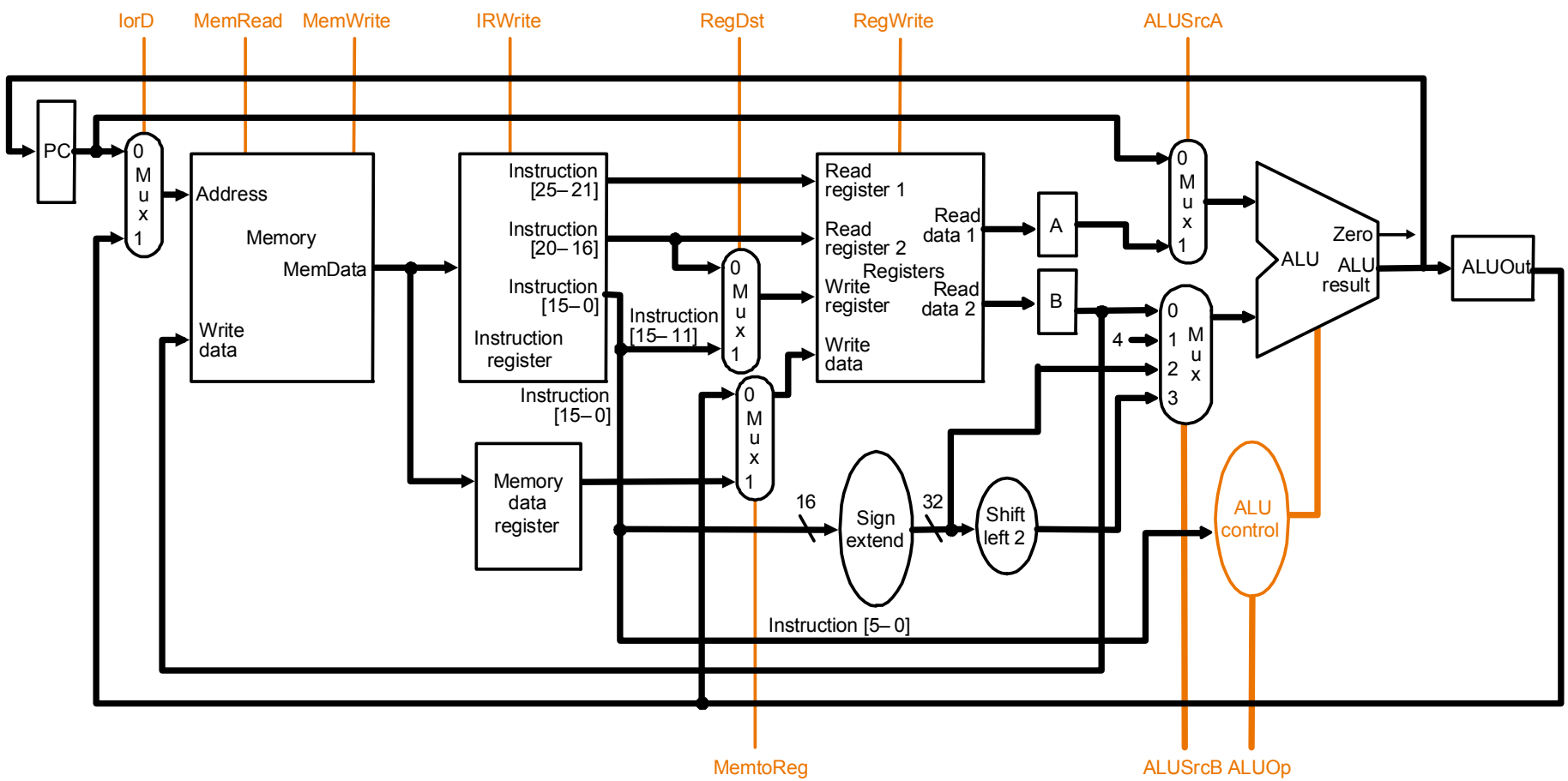
Problems With This Implementation

- This is a "single cycle" implementation
- One instruction per clock cycle
  - Therefore, clock cycle must be at least as long as the time to cover the longest path in the machine
    - This will probably be a load instruction
  - Many instructions can probably execute in a shorter time
- There is duplicated hardware
  - Multiple ALUs
- Data memory and instruction memory must be separate
  - So that they can both read (and written, if data) during the single clock cycle

Using Multiple Clocks

- Instructions can be broken in a series of smaller steps (some instructions need more than others)
- We can execute one step per clock cycle
- We will need to buffer data in registers between clock cycles / steps
- We can however re-use other hardware (e.g. ALU)
  - But will need multiplexors to route the output
- We can also use a single memory for both data and instructions
  - Instruction read and data read (or write) will be in separate "steps" of the instruction

A Multi-Cycle Implementation



Summary

- **Review Computer Architecture I !**
  - **Especially the MIPS instruction set and assembly language programming**
  - **The exam questions assume you know this stuff!**

- **Datapaths are how processor instructions are realised in hardware**
  - **The simplest implementation uses a single (long) clock cycle per instruction and requires duplicated hardware**
  - **Multi-cycle implementations allow the use of much shorter clock cycles, and short instructions to execute in a smaller number of these shorter cycles**

Next Lecture

- **On Wednesday, HLT2**
- **Introduction to pipelining**