Computer Systems Lecture 13

Motivating a Multi-Cycle Processor

Single-cycle processors aren’t good enough, their cycle time must be long enough for the most complex instruction to complete, even though the average instruction needs less time and functional units can’t be re-used within one instruction’s execution (for example, the ALU can only be used once).

Measuring Processor Speed

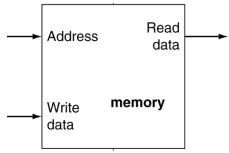
Execution time is the instruction count \* cycles per instruction \* cycle time

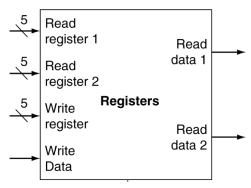
Multi-Cycle Processor: Basic Idea

Break up the execution of each instruction into multiple cycles. Ensure that the actions performed within each cycle are ‘generic’ (common to many instructions). Reuse a common set of datapath and control components across cycles.

The end result of this is that no instruction takes more time that necessary.

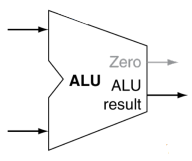
Multi-Cycle Processor: Datapath Building Blocks

One memory

* Shared between instructions and data
* Common interface

Registers

* Read early in instruction execution
* Written late (if ever) in instruction execution

One ALU

* All PC calculations (conditional and unconditional)
* All arithmetic (including branch condition evaluation)

Multi-Cycle Processor: Basic Operation

Each instruction takes multiple cycles to execute:

* Each cycle is at least one basic function performed (fetch or read registers for example)
* Multiple functions can be performed in one cycle if they use different functional units (fetch uses memory and PC+4 uses ALU for example).

An example of this would be **sw:**

* Cycle 1: Fetch (access memory)
* Cycle 2: Read registers
* Cycle 3: Compute address (use ALU)
* Cycle 4: Perform store (access memory)

Multi-Cycle Processor: Details

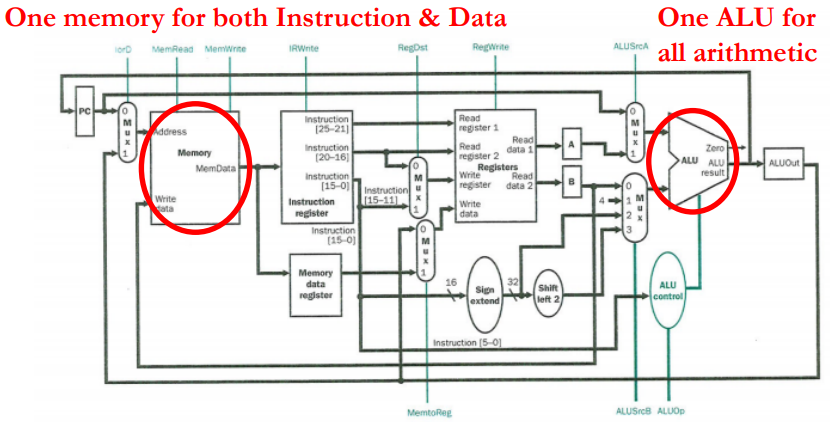
The cycle time is determined by the propagation delay through the slowest functional unit

The number of cycles varies for different instructions

At the end of each cycle, the data required in subsequent cycles must be stored somewhere

* Data used by subsequent instruction stored in memory and registers (same as a single-cycle processor)
* Data for the current instruction stored in special registers not visible to the programmer

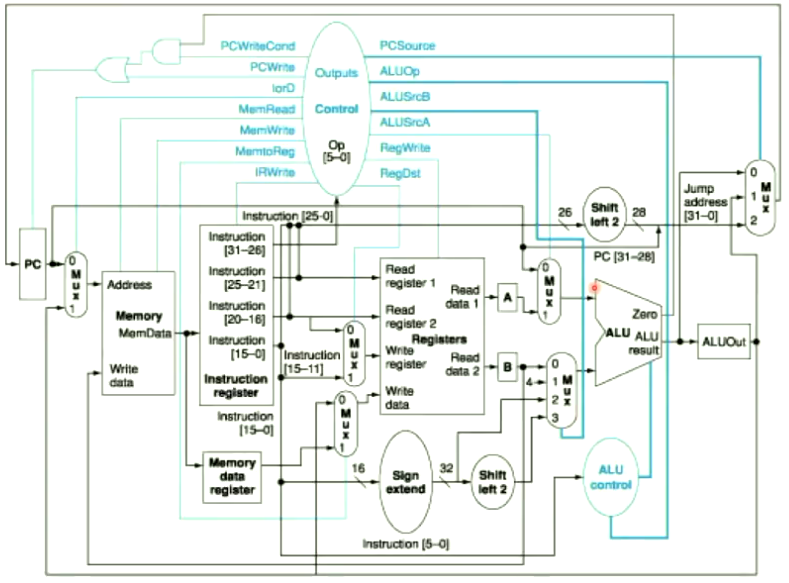
Multicycle Datapath Overview



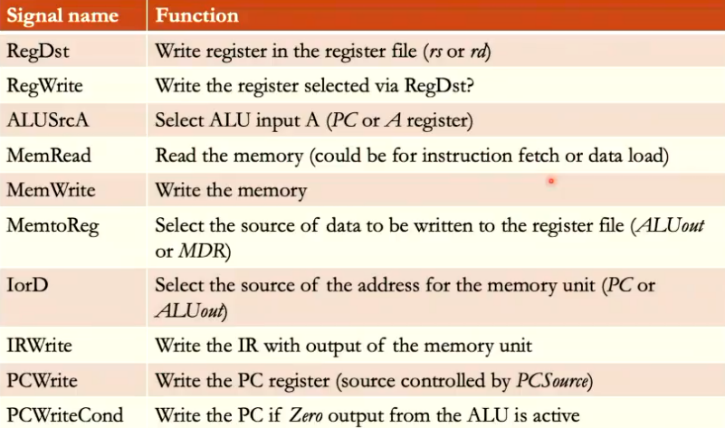
The important things to notice in this new datapath are:

* There is now only one memory space as opposed to the two (instruction and data memory) and we only have one ALU that will do all of the work that the three did before.
* We now have a mux that determines whether the next address we’re getting is from the pc (for instructions) or the ALUOut (for data).
* We now have an instruction register that allows us to view the different sections of the instruction separately.
* We have read registers A and B that held the two operands from the normal registers we’re using for the operation.
* We have two muxes that go into the ALU, the first (top) chooses between using an operand or the current instruction address and the second (bottom) chooses between the second operand, the value 4 (for incrementing the program counter), the sign-extended immediate from the instruction or the same sign-extended immediate shifted left by 2 (for the case of a branch, to make it word aligned).
* We have a register that stores the last ALU value, so it can be used in the next cycle if needed.
* We also have a memory data register that stores the last thing read from memory, and a mux that chooses whether we save that value into a usable register or save the ALUOut to a usable register.

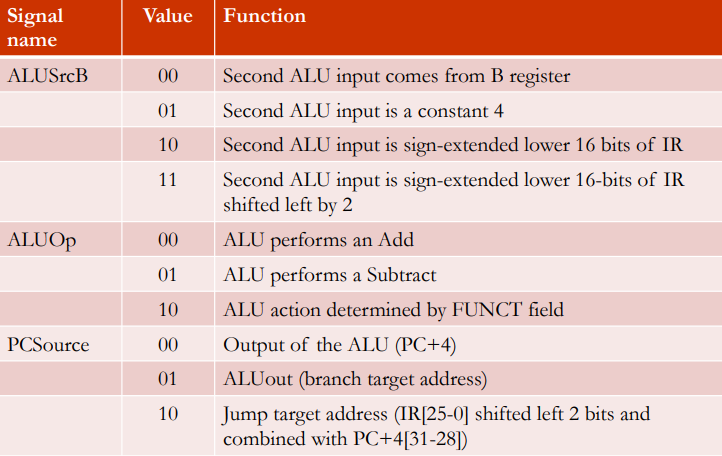
With Control

All the registers, memories and muxes need controls, we’ll also see that the control signals will need to change for each instruction depending on what cycle we’re in within the instruction.

Multi-Cycle Processor Control Signals

(1-bit)

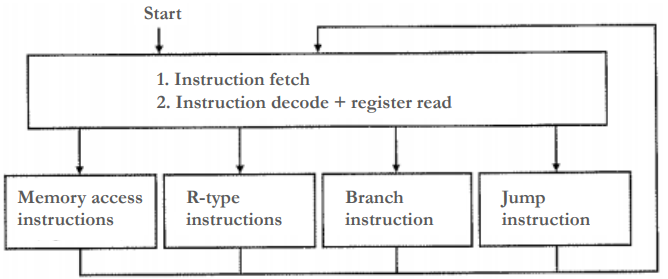
Here from all the signals from IorD down are new.  
(2-bit)



How to Design The Control

The control unit of a multicycle processor is an FSM. For a given instruction type,the sequence of control signals is determined on a cycle-by-cycle basis

Control FSM Overview



Fetch and decode is common to all instructions, no matter the type, this takes 2 cycles, 1 for fetch, 1 for decode.

The rest depends on the instruction type, this can take from 1 to 3 additional cycles.