Computer Systems Lecture 12

Processor Design – Single Cycle

There are two main parts:

* The datapath
  + Performs the data operations as commanded by the program instructions
* The control
  + Controls the datapath, memory and I/O according to the program instructions

Both of these parts use combinational and sequential circuits.

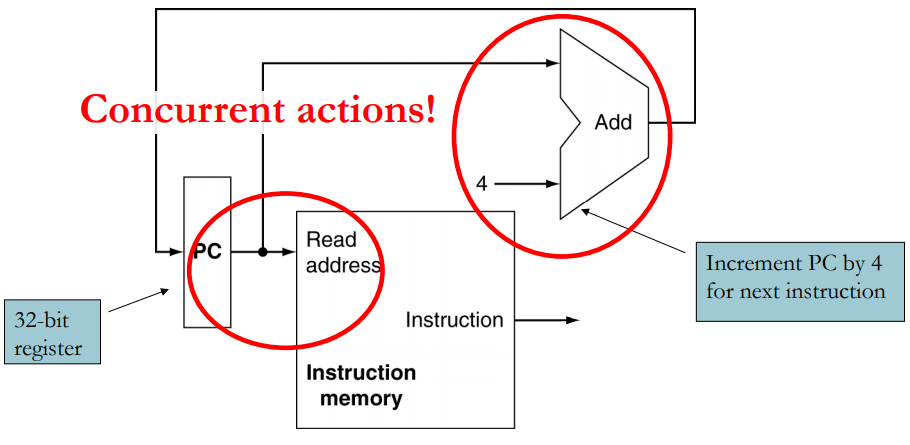
Design Steps

1. Determine the components required by understanding main processor functions
2. Build the datapath
3. Build the control

Main Processor Functions

* Fetch instruction from instruction memory
* Read the register operands
* Use the ALU for computation (arithmetic, memory address, branch target address…)
* Access data memory for load/store
* Store the result of computation or loaded data into the destination register
* Update the Program Counter (PC)

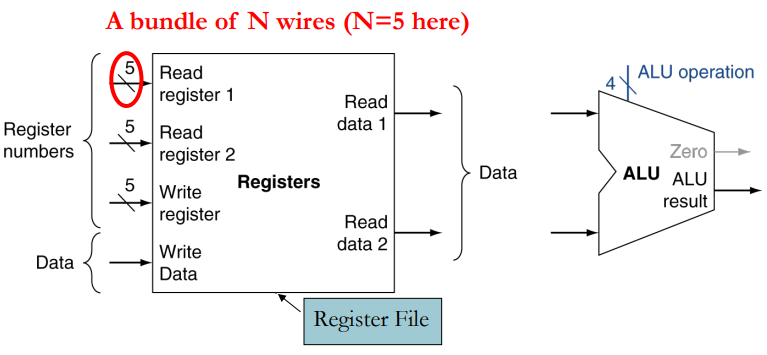
Instruction Fetch: Common to all Instructions



You fetch the instruction pointed to by the program counter from memory and then increment the counter by 4 (as each instruction is a word).

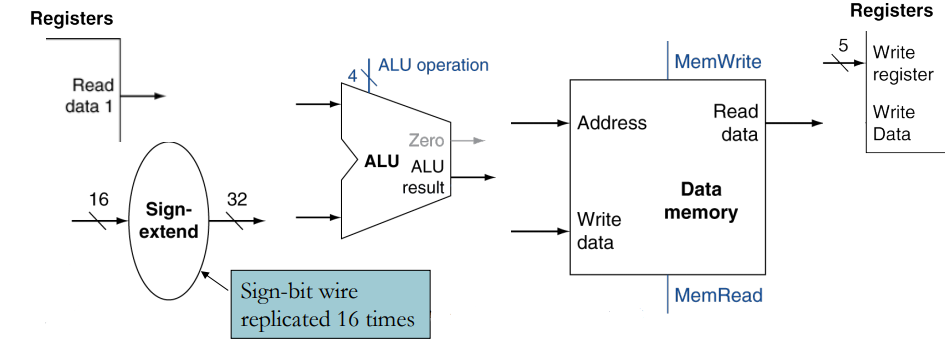
R-Format Instructions

1. Read two register operands
2. Perform arithmetic/logical operation
3. Write register result



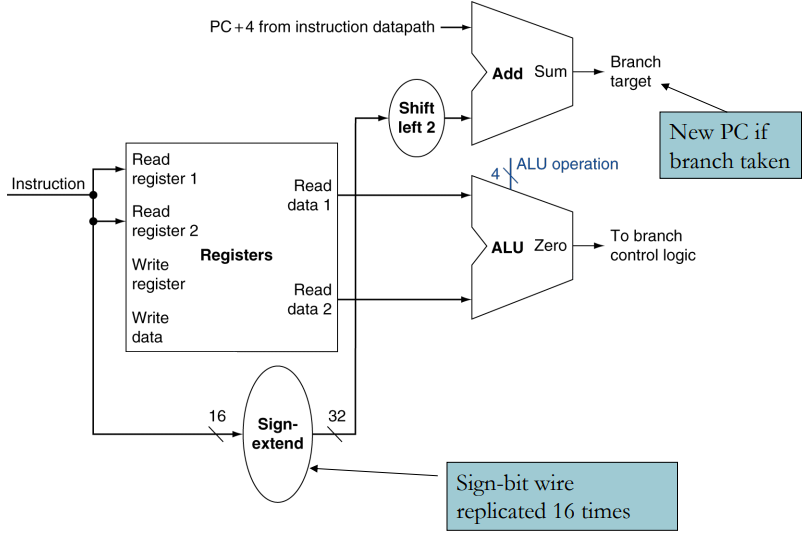
Load/Store Instructions

1. Read register operands
2. Calculate address using 16-bit offset
   1. Use the ALU but first sign-extend the offset
3. Read (for load) or write (for store) the memory
4. Load only: update destination register

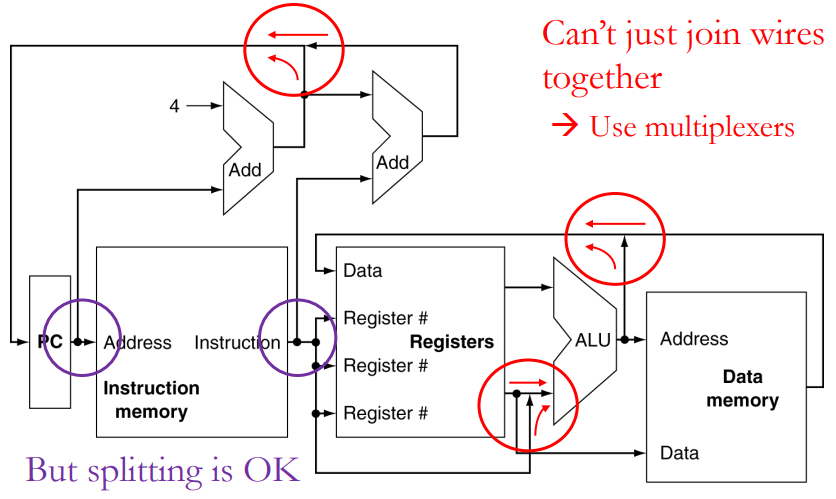


Branch Instructions (BEQ, BNE)

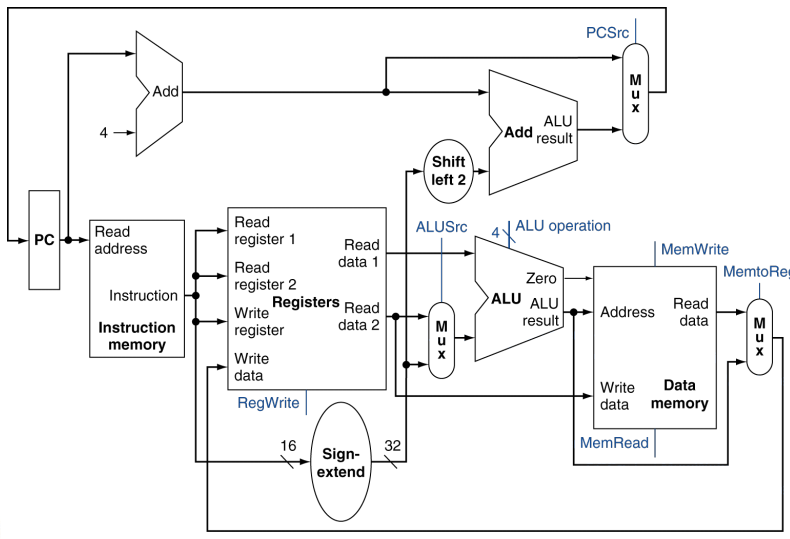
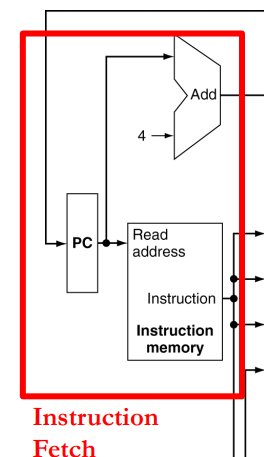
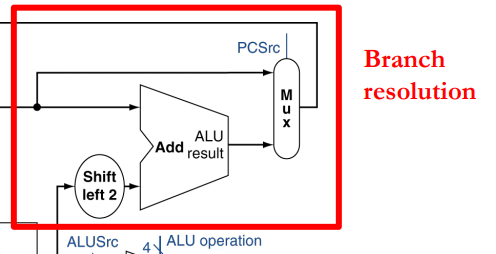
1. Read register operands
2. Compare operands
   1. Use ALU, subtract and check ALU’s Zero output
3. Calculate target address
   1. Sign-extend the immediate (offset)
   2. Shift left 2 places (word align)
   3. Add to PC+4
      1. Already calculated by instruction fetch

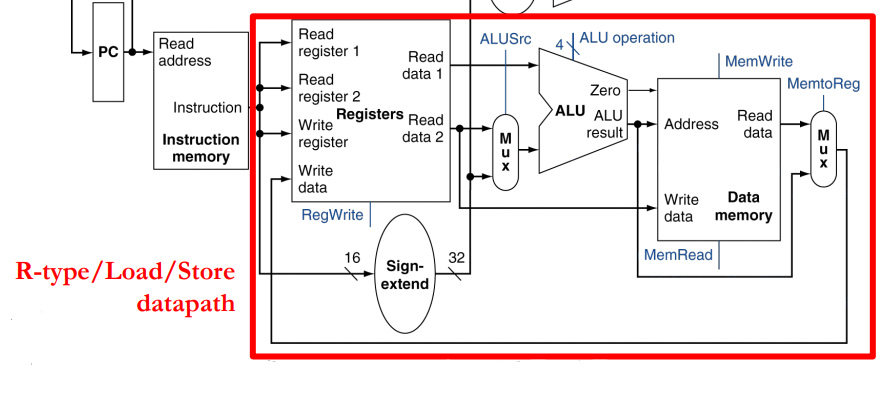


Putting It Together: Simplified Datapath



Full Datapath



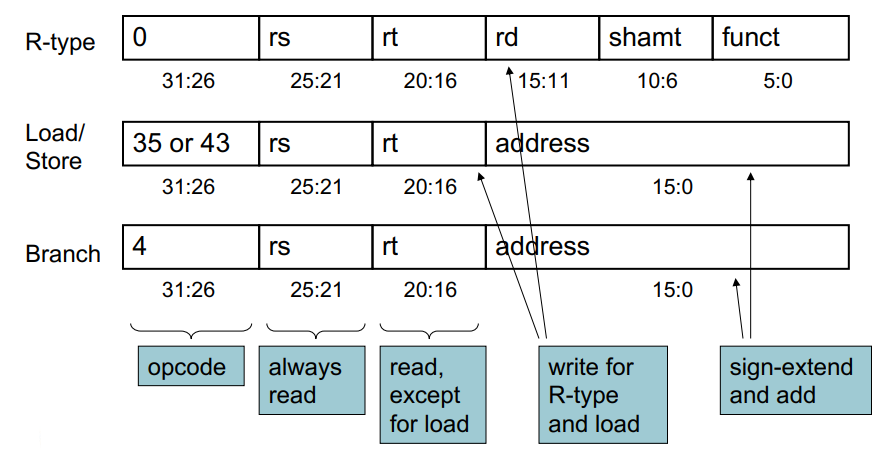


How to Design The Control Part

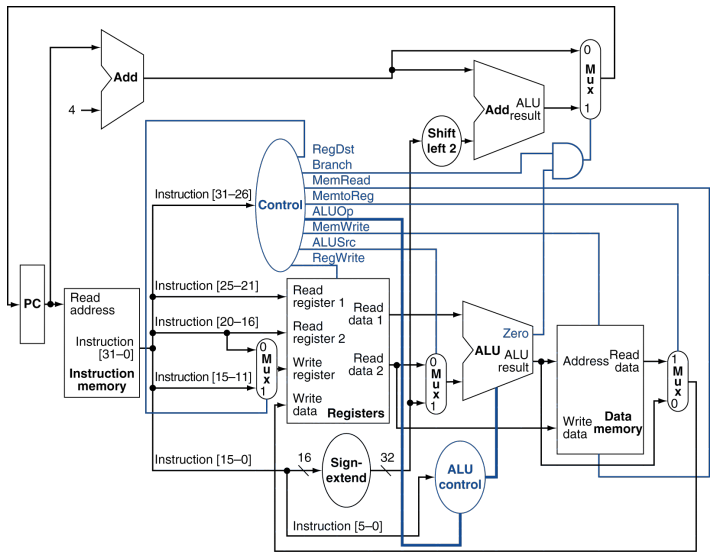
* For all control signals determine which value selects what operation, input etc…
* Make truth table of control signal values for each instruction, or instruction group
* Convert table to combinational circuit

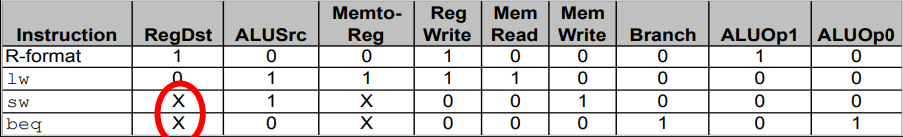
Designing the Main Control Unit

Control signals are derived from the instruction fields:



Datapath With Control

Notice how there is a main control and an ALU control.

Control Truth Table

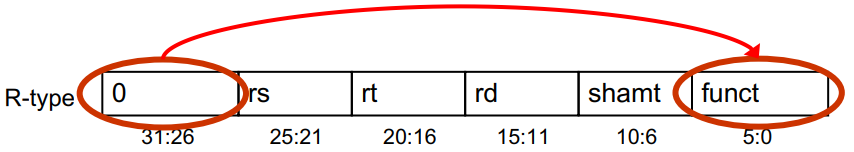
This would also have other instructions

ALU Control

Operations that use the ALU:

* Data transfers (ld/st) (always add)
* Branches (always subtract)
* All other -determined by funct field

This are derived from the opcode (operation code) from the main control logic

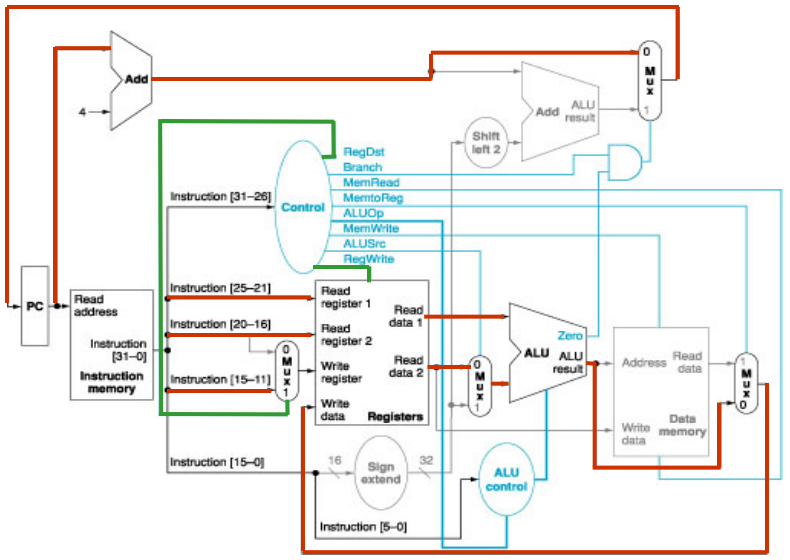


ALU control is hierarchical:

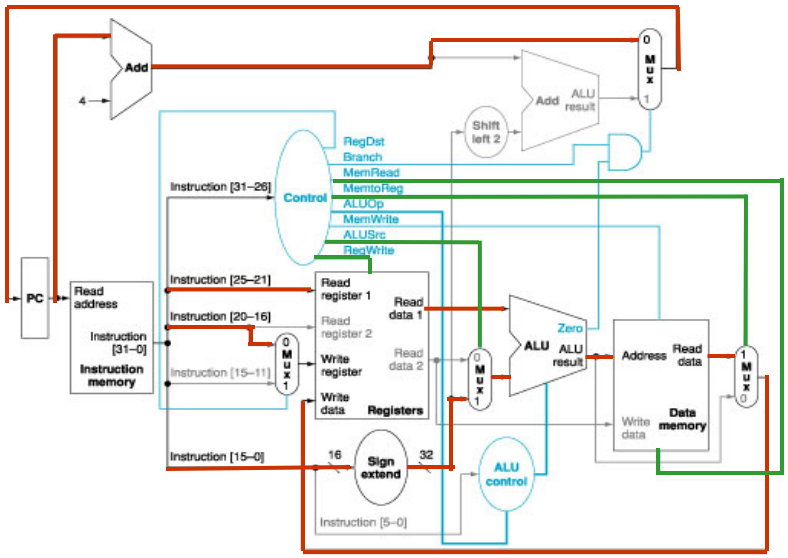
* Main control specifies which of the 3 op types (add, sub or based on funct bits). This info comes from the main control logic (ALUOp1 and ALUOp2).
* The second level provides actual ALU control signals. This comes from the bits of the instruction passed in (instruction[5-0]).

Here are some diagrams of different types of execution (red means 0 green means 1):

R-Type:



lw:



beq:

