

CSE 161 – Design and Architecture of Computer Systems

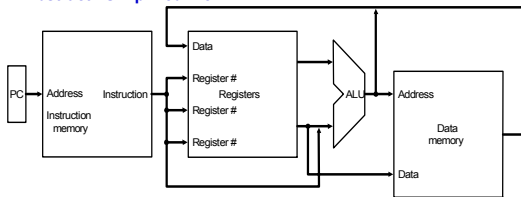
Chapter Five

The Processor: Datapath & Control

- We're ready to look at an implementation of the MIPS
- Simplified to contain only:
 - memory-reference instructions: lw, sw
 - arithmetic-logical instructions: add, sub, and, or, slt
 - control flow instructions: beq, j
- Generic Implementation:
 1. use the program counter (PC) to supply instruction address
 2. get the instruction from memory
 3. read registers
 4. use the instruction to decide exactly what to do
- All instructions use the ALU after reading the registers
Why? memory-reference? arithmetic? control flow?

More Implementation Details

□ Abstract / Simplified View:



□ Two types of functional units:

- elements that operate on data values (combinational)
 - ALU, multiplier
- elements that contain state (sequential)
 - Instruction/data memories, registers

State Elements

- Unclocked vs. Clocked
- Clocks used in synchronous logic
 - When should an element that contains state be updated?
 - Edge-triggered clocking – state elements all update their internal storage on clock edges

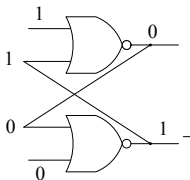


- Any combinational logic must have its inputs coming from a set of state ele. and its outputs written into a set of state ele.

An unlocked state element

□ The set-reset latch

- output depends on present inputs and also on past inputs



S-R latch: S and R can not be simultaneously asserted

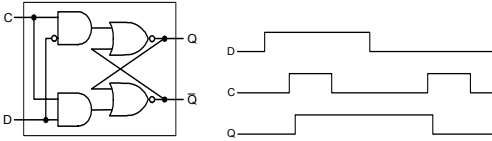
- Output is equal to the stored value inside the element (don't need to ask for permission to look at the value)

Clocked State Element – Latches and Flip-flops: simplest mem. elements

- Change of state (value) is based on the clock
- Latches: state is changed whenever the inputs change, and the clock is asserted
- Flip-flop: state changes only on a clock edge (edge-triggered methodology). Built from latches.

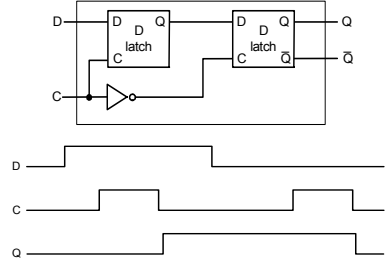
D-latch

- **Two inputs:**
 - the data value to be stored (D)
 - the clock signal (C) indicating when to read & store D
- **Two outputs:**
 - the value of the internal state (Q) and its complement
- **Positive level sensitive:** when clock is high, output is equal to input (transparent); when clock is low, output is in hold mode



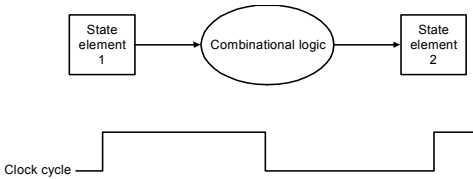
D flip-flop

- **Negative edge-triggered flip-flop** (falling edge in this case): the output is the input right before the falling edge of the clock.



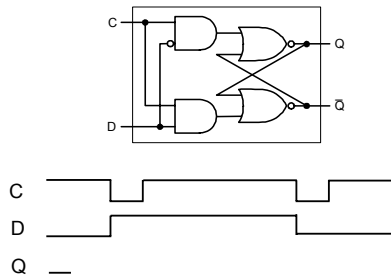
Our Implementation

- **An edge triggered methodology**
- **Typical execution:**
 - read contents of some state elements,
 - send values through some combinational logic
 - write results to one or more state elements



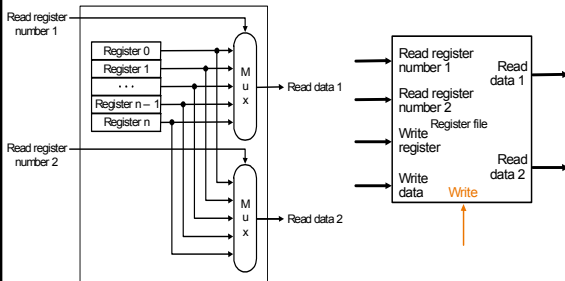
Exercise

- **Draw the timing diagram of Q for a D-latch**



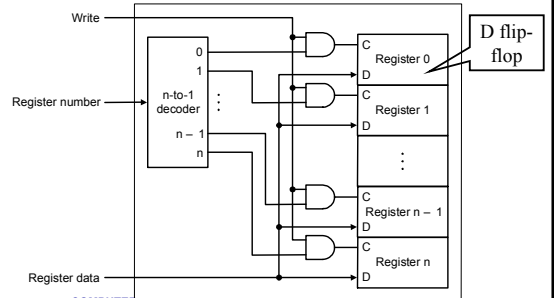
Register File

- **Built using D flip-flops**



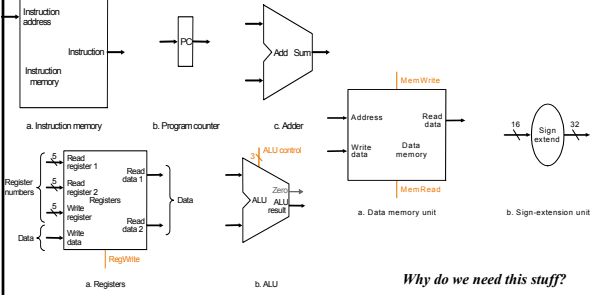
Register File

- **Note: we still use the real clock to determine when to write**



Simple Implementation

□ Include the functional units we need for each instruction



Building the Datapath

□ Use multiplexors to stitch them together

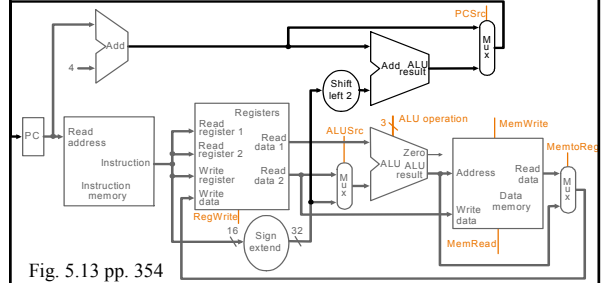
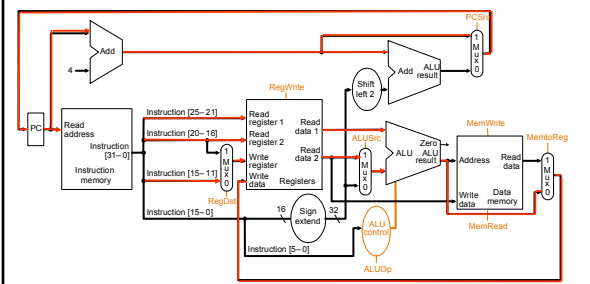
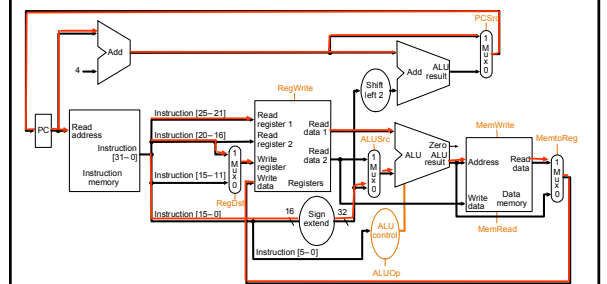


Fig. 5.13 pp. 354

R-type Instructions



Lw Instruction



Branch Instructions

