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:
- use the program counter (PC) to supply instruction address
- get the instruction from memory
- read registers
- use the instruction to decide exactly what to do

All instructions use the ALU after reading the registers

More Implementation Details

- Abstract / Simplified View:

Two types of functional units:
- elements that operate on data values (combinational)
- elements that contain state (sequential)
Major components

a. Instruction memory
   Instruction address

b. Program counter
   Instruction

   PC

   Add
   Sum

c. Adder

Data memory

MemWrite

Address

Read data

Write data

MemRead

16

Sign-extend

32

RegWrite

Read register 1

Read register 2

Write register

Read data 1

Read data 2

Registers

ALU operation

0

1

n-to-1 decoder

Register number

n - 1

n

ALU

Zero

ALU result

Read register number 1

Read register number 2

Read data 1

Read data 2

Write

0

1

C

Register 0

D

C

Register 1

D

C

Register n - 1

D

C

Register n

D
D flip-flop

- Output changes only on the clock edge

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
Fetching instructions and incrementing PC

- PC
- Read address
- Instruction
- Instruction memory
- Add
- 4
Datapath for R-type instructions

Datapath for load and store
Datapath for branching instructions

Building the Datapath
Complete Datapath