

# Levels in Processor Design

## Circuit design

- build gates (AND, OR, NAND, NOR, etc.), flip-flops, etc. from transistors

## Logic

- put gates together to form registers, adders, etc. (CSE370)

## Register transfer level

- describes the execution of instructions by showing how information is transferred and manipulated between adders, registers, etc.
  - combinational: the output is a function of the input, e.g., an adder
  - sequential: state is remembered, e.g., registers

## Architectural description

- ISA, software conventions

## System description

- memory hierarchy, I/O, number of processors, etc.

# Execution Cycle

What happens when an instruction executes

## (1) **fetch**

- sent the PC to memory
- transfer an instruction from memory to the CPU
- increment the PC

## (2) **decode** & read the ALU source

- registers
- immediate from the instruction

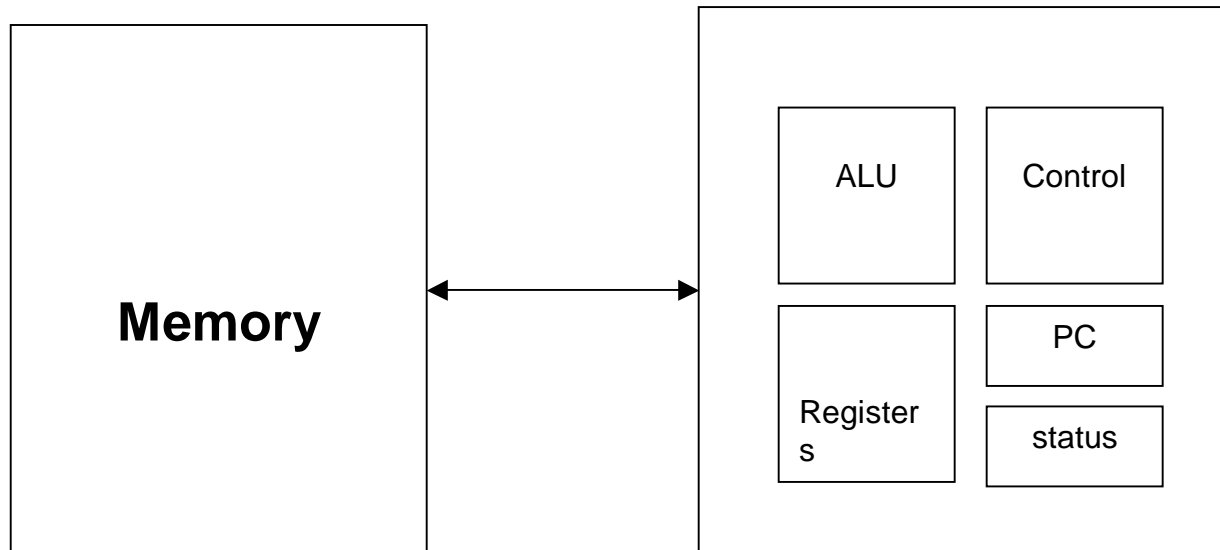
## (3) **execute**

- an ALU operation
- **effective address** calculation
  - address to memory
  - memory access
  - data to memory (if a store)
  - data from memory (if a load)
- branch target calculation
  - condition check
  - change the PC
  - save the return address on a **jal**

## (4) **store** the result

- from the ALU
- from memory

# Processor Datapath & Control Unit



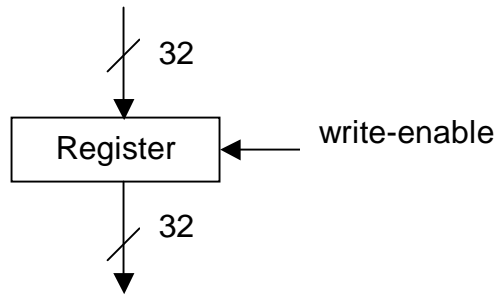
## Datapath:

- **combinational & sequential** logic
- hardware that implements the operations: ALU, incrementer
- storage for the program (**processor state**): GPRs, PC, status
- wires & buses: data flow between the components

## Control

- sends signals to the datapath elements
- specifies what operations to perform, what data to move, when to move it, where to move it

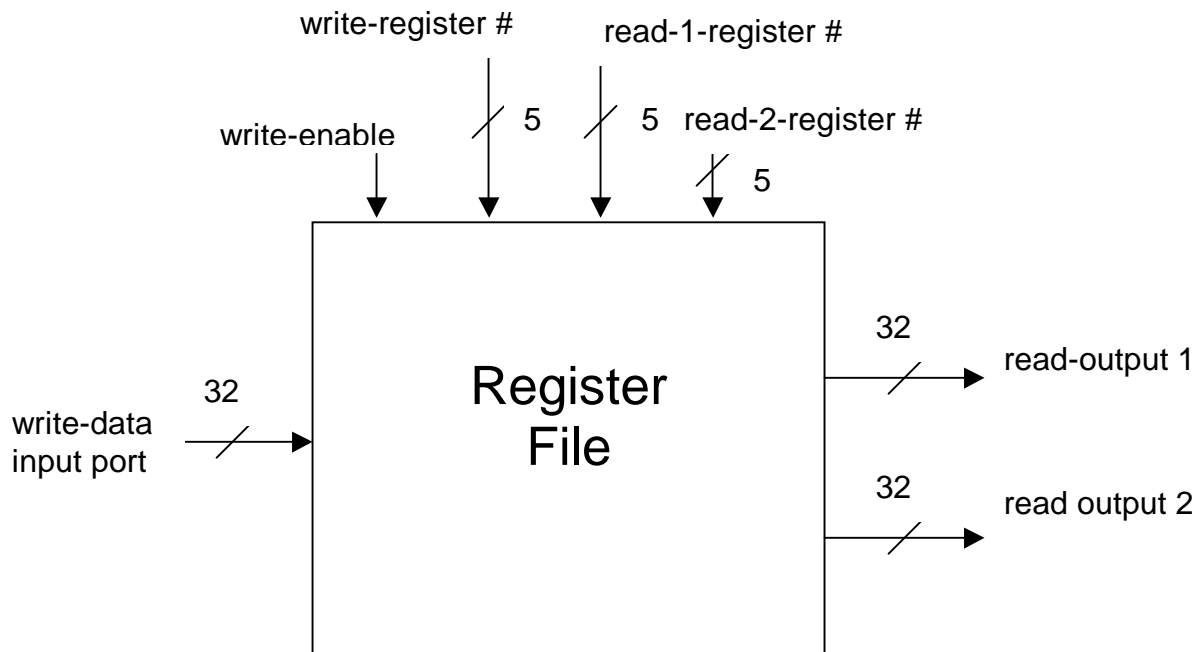
# Datapath Building Blocks: Sequential



## Register

- holds 32 bits
- can be read or written

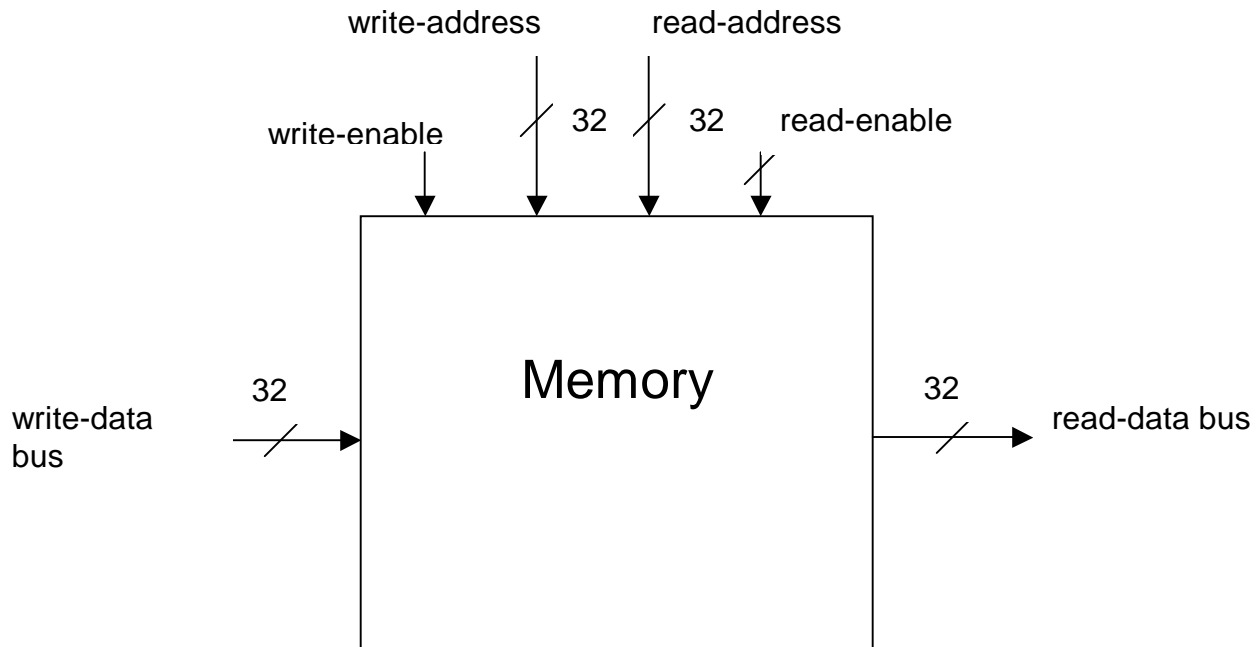
# Datapath Building Blocks: Sequential



## Register File

- always reads
- writes controlled by enable
- can both read and write in the same cycle
  - write first, read second
- can read two register values in the same cycle

# Datapath Building Block: Sequential



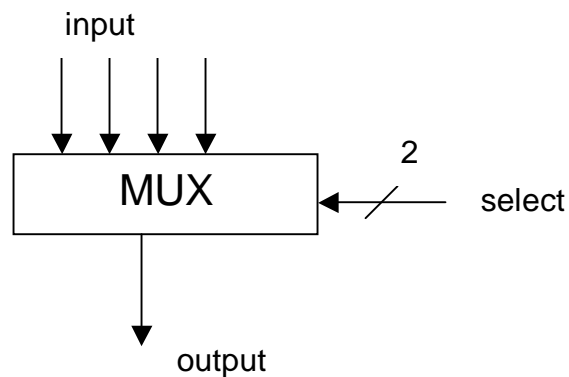
## Memory

- can only read or write one location in a cycle

# Datapath Building Block: Combinational

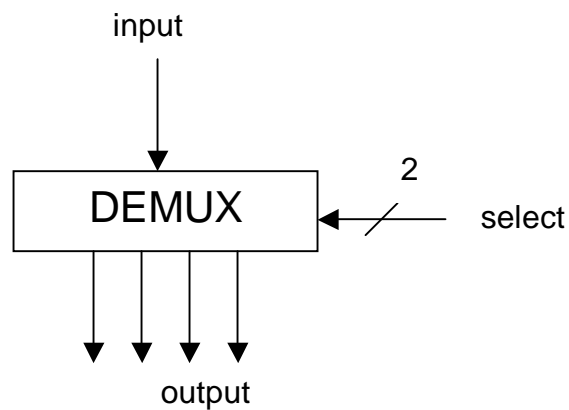
## Multiplexor (MUX):

- selects one of its inputs for output
- input selected depends on **select** signal

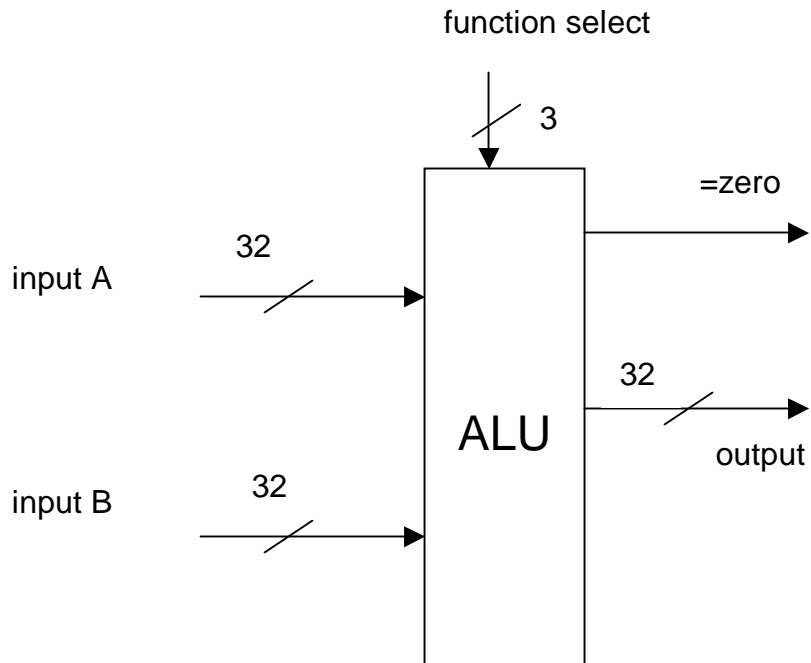


## Demultiplexor (selector)

- routes the input to one of the outputs
- the output chosen depends **select**



# Datapath Building Block: Combinational



## ALU

- Computes arithmetic & logical functions



# **Single-cycle Datapath**

No hardware can be reused by the same instruction

(memory units not really on the CPU)