Introduction

Lecture 1

Robb T. Koether

Hampden-Sydney College

Mon, Aug 26, 2019

1. Program Performance
2. The Five Components
3. From High-Level Language to Machine Language
4. The Processor
5. Assignment

Outline

1. Program Performance
2. The Five Components
3. From High-Level Language to Machine Language
4. The Processor
5. Assignment

Program Performance

- Program performance depends on
  - Algorithm
  - Programming language
  - Compiler
  - Processor's instruction set
  - Processor's clock speed
  - Memory system
  - I/O system
  - Operating system

Program Performance

- Program performance depends on
  - Algorithm
  - Programming language
  - Compiler
  - Processor's instruction set
  - Processor's clock speed
  - Memory system
  - I/O system
  - Operating system

Organizational Levels

- Organizational levels of a program
  - Programmer's software
  - System software (I/O, memory allocation, etc.)
  - Hardware

Organizational Levels

- Organizational levels of a program
  - Programmer's software
  - System software (I/O, memory allocation, etc.)
  - Hardware

Outline

1. Program Performance
2. The Five Components
3. From High-Level Language to Machine Language
4. The Processor
5. Assignment

The Five Components

- Computers consist of five main components.
  - Input devices
  - Output devices
  - Memory
  - Datapath
  - Control

The Five Components

- Computers consist of five main components.
  - Input devices
  - Output devices
  - Memory
  - Datapath
  - Control

Outline

1. Program Performance
2. The Five Components
3. From High-Level Language to Machine Language
4. The Processor
5. Assignment

High-Level Language

High-Level Language

```
swap(int v[], int k)
{
    int temp;
    temp = v[k];
    v[k] = v[k + 1];
    v[k + 1] = temp;
}
```

- The above C code is a function to swap two array elements.

High-Level Language

Assembly Language

```
swap:
        sll     $2,$5,2
        add     $2,$4,$2
        lw      $15,0($2)
        lw      $16,4($2)
        sw      $16,0($2)
        sw      $15,4($2)
        jr      $31
```

- This is the same function in MIPS assembly language.

High-Level Language

Machine Language

```
00000000000001010001000010000000
00000001000001000010000000100000
10001100010011110000000000000000
10001100010100000000000000000100
10101100010100000000000000000000
10101100010011110000000000000000
00000011111000000000000000001000
```

- This is the same function in MIPS machine language.

High-Level Language

Machine Language

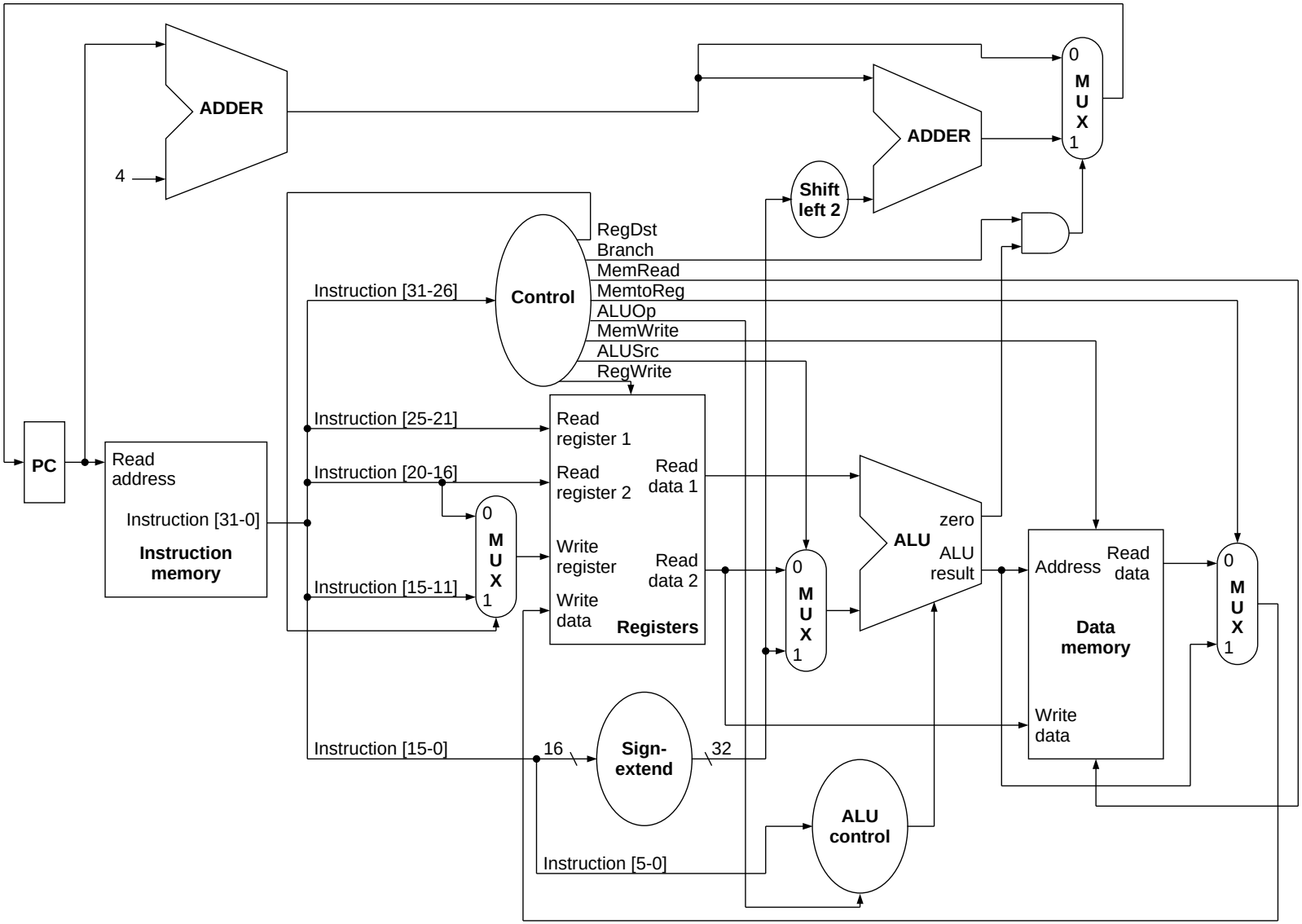
```
000000 00000 00101 00010 00010 000000
000000 00100 00010 00010 00000 100000
100011 00010 01111 00000 00000 000000
100011 00010 10000 00000 00000 000100
101011 00010 10000 00000 00000 000000
101011 00010 01111 00000 00000 000100
000000 11111 00000 00000 00000 001000
```

- This is the same function in MIPS machine language.

Outline

1. Program Performance
2. The Five Components
3. From High-Level Language to Machine Language
4. The Processor
5. Assignment

The Processor



Outline

1. Program Performance
2. The Five Components
3. From High-Level Language to Machine Language
4. The Processor
5. Assignment

Assignment

- Assignment
  - Download the MARS MIPS simulator and assemble and run the program `add2ints.asm`.