

Integer Programming

Lecture 4

Sections 2.5 - 2.6

Robb T. Koether

Hampden-Sydney College

Mon, Sep 2, 2019

- 1 Addition
 - The Add Unsigned Instructions
 - The Subtract Unsigned Instructions
- 2 Input and Output in MARS
- 3 Dialog Boxes in MARS
- 4 Assignment

Outline

- 1 Addition
 - The Add Unsigned Instructions
 - The Subtract Unsigned Instructions

- 2 Input and Output in MARS

- 3 Dialog Boxes in MARS

- 4 Assignment

Outline

- 1 Addition
 - The Add Unsigned Instructions
 - The Subtract Unsigned Instructions

2 Input and Output in MARS

3 Dialog Boxes in MARS

4 Assignment

The Add Unsigned Instructions

The `addu` and `addiu` Instructions

```
addu    rd, rs, rt           #  $rd = rs + rt$   
addiu   rt, rs, imm          #  $rt = rs + imm$ 
```

- The `addi` and `addiu` instructions will add the second and third operands and store the result in the first operand.
- All operands are interpreted as unsigned integers.
- The only difference between signed and unsigned addition is that signed addition may generate an overflow exception, while unsigned addition will not.

The Add Unsigned Instructions

Example of `addu` and `addiu` Instructions

```
add      $t0, $t1, $t2    # t0 = t1 + t2 (signed)
addu     $t0, $t1, $t2    # t0 = t1 + t2 (unsigned)
addi     $t0, $t1, 100    # t0 = t1 + 100 (signed)
addiu    $t0, $t1, 100    # t0 = t1 + 100 (unsigned)
```

```
012a4020 = 000000 01001 01010 01000 00000 100000
```

```
012a4021 = 000000 01001 01010 01000 00000 100001
```

```
21280064 = 001000 01001 01000 0000000001100100
```

```
25280064 = 001001 01001 01000 0000000001100100
```

Outline

- 1 Addition
 - The Add Unsigned Instructions
 - The Subtract Unsigned Instructions

2 Input and Output in MARS

3 Dialog Boxes in MARS

4 Assignment

The `sub` and `subu` Instruction

The `sub` and `subu` Instructions

```
sub    rd, rs, rt          #  $rd = rs - rt$   
subu   rd, rs, rt          #  $rd = rs - rt$ 
```

- The `sub` and `subu` instructions will subtract the third operand from the second operand and store the result in the first operand.
- All three operands are registers.
- The signed subtraction can generate an overflow exception, while the unsigned subtraction cannot.
- Otherwise, there is no difference.

The `sub` and `subu` Instruction

Example of `sub` and `subu` Instructions

```
sub    $t3, $t4, $t5    # t3 = t4 - t5 (signed)
subu   $t3, $t4, $t5    # t3 = t4 - t5 (unsigned)
```

```
018d5822 = 000000 01100 01101 01011 00000 100010
```

```
018d5823 = 000000 01100 01101 01011 00000 100011
```

- The `sub` and `subu` instructions will subtract the third operand from the second operand and store the result in the first operand.
- All three operands are registers.
- The signed subtraction can generate an overflow exception, while the unsigned subtraction cannot.
- Otherwise, there is no difference.

Outline

- 1 Addition
 - The Add Unsigned Instructions
 - The Subtract Unsigned Instructions

- 2 **Input and Output in MARS**

- 3 Dialog Boxes in MARS

- 4 Assignment

Input in MARS

Input in MARS

```
li      $v0,5      # System call 5
syscall          # Read an integer
move    $s0,$v0   # Store integer in $s0
```

- We will use system calls to read and write integers and strings.
- System call #5 will read one integer from the MARS console.
- The value is read into register `$v0`.

Output in MARS

Output in MARS

```
move    $a0,$s0    # Move integer to $a0
li      $v0,1      # System call 1
syscall                # Print an integer
```

- System call #1 will write one integer to the MARS console.
- We must move the value to be written to register \$a0.

Examples

- Write a program that will read two integers, add them together, and print the result.
 - Test the program with sample data.
 - Add 2147483647 and 1.
 - Add -2147483648 and -2147483648.
- Modify the program to subtract the second integer from the first.
 - Test the program with sample data.
 - Subtract 1 from -2147483648.
 - Subtract -1 from 2147483647.

Outline

- 1 Addition
 - The Add Unsigned Instructions
 - The Subtract Unsigned Instructions

- 2 Input and Output in MARS

- 3 Dialog Boxes in MARS**

- 4 Assignment

Input Dialog Boxes

Input Dialog Box

```
.data
prmt:  .ascii "Enter an integer:"
.text
la     $a0, prmt    # Load addr of prompt
li     $v0, 51      # syscall 51 = input box
syscall                    # Open dialog box
move   $s0, $a0     # Save the integer in s0
```

- syscall 51 will open a dialog box allowing the user to input an integer.
- The user enters an integer and then clicks either “OK” or “Cancel.”

Input Dialog Boxes

- Upon return, the integer is in register `$a0` and the status is in register `$a1`.
- Return status

0 = OK.

-1 = Error.

-2 = User clicked "OK."

-3 = User clicked "Cancel."

Output Dialog Boxes

Output Dialog Box

```
.data
msg:    .asciiz "The answer is "
.text
la      $a0, msg      # Load addr of msg
move    $a1, $s0      # Move integer to a1
li      $v0, 56        # syscall 56 = output box
syscall                # Open dialog box
```

- syscall 56 will open a dialog box allowing the user to output an integer.
- The integer to be printed is in \$a1.
- There is no return value.

Outline

- 1 Addition
 - The Add Unsigned Instructions
 - The Subtract Unsigned Instructions

- 2 Input and Output in MARS

- 3 Dialog Boxes in MARS

- 4 Assignment**

Assignment

Assignment

- Read Sections 2.5 - 2.6.