## Logic Structures in Assembly Language

XD

Peking University

2021

### Logic Structures in Assembly Language

ΧD

### ommand

Program Counter Condition Codes Compare

Test

Conditions

set

lumn Tables

### Conditional

Conditional Jump

### Logic Structures

Conditional Branches (if) Loops (for, while)

Switch Statements

ercises

## Outline

### Commands

Program Counter

Condition Codes

Compare

Test

Conditions

Set

Jumps

Jump Tables

### Conditional Commands

Conditional Jumps

Conditional Moves

## Logic Structures

Conditional Branches (if)

Loops (for, while)

Switch Statements (switch)

### **Exercises**

### Logic Structures in Assembly Language

XD

### ommands

Program Count Condition Code

ompare

rest

Conditions

et

Jump Tabler

Jump Tables

### Command

Conditional Jumps

### ogic Structures

Conditional Branches (if

Switch Statements

xercises

Exercises

4 ロ ト 4 倒 ト 4 亘 ト 亘 め 9 0 0

## Basic Assembly Commands

In Lesson 5, we learnt a bunch of new features, syntax and commands. Now let's take a little time to go over them once again.

### Logic Structures in Assembly Language

ΧD

### Commands

Program Counte Condition Codes

Compar

Condition

Set

Jumps

Jump Tables

## onditional

Conditional Jump

### agic Structures

Conditional Branches (if)

witch Statements

Definition: A register storing the address of the next command.

### Logic Structures in Assembly Language

XD

### Command

## Program Counter

Compare

Conditions

Conditions

Jumps

Jump Tables

### Condition

Conditional Jump

### agic Structures

Conditional Branches (if

Switch Statements

Exercises

- Definition: A register storing the address of the next command.
- Only an abstract concept; Has different implementations, like %rip(RIP) in x86.

### Logic Structures in Assembly Language

XD

### Command

### Program Counter

Compare

Conditions

Londitions

Set

Jump Tables

### Comman

Commands
Conditional Jum
Conditional Mov

### Logic Structures

Conditional Branches (if) Loops (for, while)

Evercises

- Definition: A register storing the address of the next command.
- Only an abstract concept; Has different implementations, like %rip(RIP) in x86.
- Cannot be directly written (read-only).

Logic Structures in Assembly Language

XD

Program Counter

- Definition: A register storing the address of the next command.
- Only an abstract concept; Has different implementations, like %rip(RIP) in x86.
- Cannot be directly written (read-only).

### Remark

Can be used to access data stored in memory with relative address difference.

# Example movg 8(%rin)

movq 8(%rip) %rax

Logic Structures in Assembly Language

XD

Commands

Program Counter Condition Codes

Conditions

Set

Jump Tables

Commands

Conditional Jumps Conditional Moves

ogic Structures

Conditional Branches (if) Loops (for, while) Switch Statements (switch

## **Condition Codes**

► Synonym: State Flag

### Logic Structures in Assembly Language

XD

### ommand:

Program Counter

### Condition Codes

Compare

Test

Conditions

Set

lumn Tables

### Condition

ommands onditional Jump

### . . .

Logic Structures

Loops (for, while)

Switch Statements (

## **Condition Codes**

► Synonym: State Flag

► Four Condition Codes:

Tour Condition Codes.		
	Full Name	Meaning
CF	Carry Flag	unsigned overflow
ZF	Zero Flag	zero
SF	Sign Flag	negative result
OF	Overflow Flag	signed overflow

### Logic Structures in Assembly Language

XD

Commands

Program Counter

Condition Codes

mpare

Test

Conditions

Set

iumps i min

Jump Tables

Commands

Conditional Jumps Conditional Moves

ogic Structures

Conditional Branches (if)
Loops (for, while)

Evercises

## **Condition Codes**

► Synonym: State Flag

► Four Condition Codes:

Tour condition codes.		
	Full Name	Meaning
CF	Carry Flag	unsigned overflow
ZF	Zero Flag	zero
SF	Sign Flag	negative result
OF	Overflow Flag	signed overflow

 Programmers should know which sort of (signed, unsigned) data they are dealing with. No checker inside CPU. Logic Structures in Assembly Language

XD

ommands

Program Counter

Condition Codes

Lompan Fest

Conditions

et

Jumps L.... Tablas

Conditiona

Commands
Conditional Jump

ogic Structures

Conditional Branches (if)
Loops (for, while)
Switch Statements (switch

vercises



## How condition codes are set?

The state of condition code depends on the result of the last arithmetic command.

### Logic Structures in Assembly Language

XD

### Condition Codes

## How condition codes are set?

Logic Structures in Assembly Language

XD

Condition Codes

The state of condition code depends on the result of the last arithmetic command.

## Example

Suppose all the flags are 0 at first, doing sub %rax %rax mov %rbx %rax only changes ZF to be 1.

## How condition codes are set?

Logic Structures in Assembly Language

ΧD

mmands

Program Counter

Condition Codes

Test

Condition

Set

lumn Tables

Conditional

Commands
Conditional Jump

Conditional Move

Logic Structures

Conditional Branches (if) Loops (for, while)

Switch Statement

Evercises

The state of condition code depends on the result of the last arithmetic command.

## Example

Suppose all the flags are 0 at first, doing sub %rax %rax mov %rbx %rax only changes ZF to be 1.

### Remark

leaq doesn't change the condition code.

Set

Jumps

Jump Tables

Commands

Conditional Jump

Conditional Moves

### Logic Structures

Conditional Branches (if Loops (for, while)

Switch Statemen

Exercises

The state of condition code depends on the result of the last arithmetic command.

## Example

Suppose all the flags are 0 at first, doing sub %rax %rax mov %rbx %rax only changes ZF to be 1.

## Remark

leaq doesn't change the condition code.

### Remark

inc and dec set OF and ZF flags, but leave CF untouched.

## Compare

There are two commands designed explicitly for setting the condition code: cmp (compare) and test.

### definition

cmpX S2 S1 ( $X \in \{b, w, l, q\}$ ): Do S1 - S2 without storing the result.

Logic Structures in Assembly Language

XD

Command

Program Counte

Compare

est

onditions

Set

Jump Tables

Condition

Commands
Conditional Jump

### Logic Structures

Conditional Branches (if)
Loops (for, while)
Switch Statements (switch

ercises

## Compare

There are two commands designed explicitly for setting the condition code: cmp (compare) and test.

### definition

cmpX S2 S1 ( $X \in \{b, w, l, q\}$ ): Do S1 - S2 without storing the result.

### Remark

Used to compare the order of two (un)signed numbers.

Logic Structures in Assembly Language

XD

Commands

Program Counte Condition Codes

Compare

est

onditions

et

lumn Tables

Conditional Commands

Conditional Jumps Conditional Moves

Logic Structures

Conditional Branches (if)
Loops (for, while)

## Test

## definition

testX S2 S1 ( $X \in \{b, w, l, q\}$ ): Do S1 & S2 without storing the result.

### Logic Structures in Assembly Language

XD

### Tost

## Test

## definition

testX S2 S1 ( $X \in \{b, w, l, q\}$ ): Do S1 & S2 without storing the result.

### Remark

Useful with a mask to check if some given digits are all 0.

Logic Structures in Assembly Language

XD

Commands

Compare

Test

Conditions

set .

Jump Tables

Conditional

Commands
Conditional Jump

La min Churchung

Logic Structures

Conditional Branches (if)
Loops (for, while)
Switch Statements (switch

## Test

# definition

testX S2 S1 ( $X \in \{b, w, l, q\}$ ): Do S1 & S2 without storing the result.

### Remark

Useful with a mask to check if some given digits are all 0.

### Remark

Frequently used as testq %rax %rax to check if a number if zero. (Shorter than cmpq %rax 0!)

Logic Structures in Assembly Language

XD

Commands

Program Counte Condition Codes Compare

Test

Conditions

Jumps

Jump Tables

Commands
Conditional Jump

Conditional Jumps Conditional Moves

Logic Structures

Conditional Branches (if)
Loops (for, while)
Switch Statements (switch

### Canonical

n: not

e: equal ZF z: zero ZF

Logic Structures in Assembly Language

XD

### Conditions

- Canonical
  - n: not
  - e: equal ZF
  - z: zero ZF
- Signed
  - ▶ g: greater (SF ⊕ OF)&ZF
  - ► I: less SF ⊕ OF

### Logic Structures in Assembly Language

XD

### Commands

Program Counter Condition Codes Compare

### Conditions

Jumps

Jump Tables

## Comman

Conditional Jump

### Logic Structures

Conditional Branches (if)

Switch Statements

- Canonical
  - n: not
  - e: equal ZF
  - z: zero ZF
- Signed
  - ▶ g: greater (SF ⊕ OF)&ZF
  - ► I: less SF ⊕ OF
- Unsigned
  - a: above CF& ZF
  - b: below CF

### Logic Structures in Assembly Language

XD

### Conditions

Canonical

n: not

e: equal ZF

> z: zero 7F

Signed

▶ g: greater (SF ⊕ OF)&ZF

I: less SF ⊕ OF

Unsigned

► a: above CF& 7F

b: below CF

### Remark

n is used as a prefix to negate the conditions that follow.

## Remark

e can be used as a suffix to a,b,g,l representing 'or equal'.

Logic Structures in Assembly Language

XD

Conditions

ompare

Condition

.

Set

Jumps Tables

Jump Tables

Commands

Conditional Jumps Conditional Move

Logic Structures

Conditional Branches (

Loops (for, while)

We need a way to retrieve and manipulate the condition codes.

### definition

set X D (X is a condition): moving the corresponding combination of condition codes into the lower-order byte of D.

## Remark

Set doesn't clear the high-order bytes.

## **Jumps**

Logic Structures in Assembly Language

XD

ommands

Program Counte Condition Codes

ort

onditions

Set

Jumps

Jump Tables

Conditional Commands Conditional Jump

Conditional Jump Conditional Move

Logic Structures

Conditional Branches (if)
Loops (for, while)
Switch Statements (switch

Evercises

To implement more complicated structures while minimizing the size of the assembly program, jumps are introduced.

### definition

jmp X: jump to an memory address specified by X (can be a label)

jmp \*R: jump to the memory address specified in register R jmp \*(R): jump to the memory address specified in the memory specified by register R

est

Conditions

Jumps

Jump Tables

Conditional lum

Conditional Jump Conditional Move

Logic Structures

Conditional Branches (if)
Loops (for, while)

\_ .

To implement more complicated structures while minimizing the size of the assembly program, jumps are introduced.

### definition

jmp X: jump to an memory address specified by X (can be a label)

jmp \*R: jump to the memory address specified in register R jmp \*(R): jump to the memory address specified in the memory specified by register R

## Remark

PC-relative encoding have two advantages:

- more compactly encoded instruction
- more portable code

## Jump Tables

We can also use a constant table to refer to what's the jump destination.

### definition

An anchor label and an alignment specification.

### Logic Structures in Assembly Language

XD

### Jump Tables

Jump Tables

We can also use a constant table to refer to what's the jump

## usage

imp .L2(,%rax,8): jump to the label specified by %rax

.section .rodata /\* stands for read-only data \*/

.L4: /\* the anchor label \*/

.align 8 /\* specifies the alignment used in the table \*/

### 4 ロ ト 4 倒 ト 4 亘 ト 亘 め 9 0 0

## destination

definition

An anchor label and an alignment specification.

structure

.quad .L8 .quad .L2

## Jump Tables

We can also use a constant table to refer to what's the jump destination.

### definition

An anchor label and an alignment specification.

### structure

```
.section .rodata /* stands for read-only data */
.align 8 /* specifies the alignment used in the table */
.L4: /* the anchor label */
.quad .L8
.quad .L2
```

. . .

### usage

jmp .L2(,%rax,8): jump to the label specified by %rax

### Remark

Jump tables help perform jumps automatically.

Logic Structures in Assembly Language

XD

### ommands

Program Counte

est

Conditions

Jumps

Jump Tables

Conditional Commands

Conditional Jum

Conditional Moves

### Logic Structures

Conditional Branches (if)
Loops (for, while)
Switch Statements (switch

## Conditional Jumps

Logic Structures in Assembly Language

XD

Conditional lumps

To harness the power of condition codes, conditional jumps are introduced.

### definition

jX D (X is a condition): jump to the address specified by D if X is satisfied.

### Remark

Conditional jump are the reason why we could implement logic structure.

## rep&repz

In the assembly code provided in the CSAPP textbook and the bomb lab, sometimes **ret** is written as **rep ret** or **repz ret**.

So, why are they there in the first place?

Logic Structures in Assembly Language

XD

ommands

Program Count
Condition Code

Гest

Conditions

Sot

Jumps

Jump Tubics

Conditional

Conditional lumps

onditional Jumps.

Conditional Moves

Logic Structures

Conditional Branches (if

oops (for, while)

vercises

## rep&repz

Logic Structures in Assembly Language

XD

Conditional lumps

In the assembly code provided in the CSAPP textbook and the bomb lab, sometimes ret is written as rep ret or repz ret

So, why are they there in the first place?

### Answer

They are only used for AMD CPUs since these CPUs cannot handle the return address if ret is reached from a conditional jump. In another word, rep and repz are meaningless occupiers that could avoid this situation from happening.

## Condition Moves

Before diving into details why we introduce conditional moves, we first review the definition of conditional moves(cmov).

### definition

cmovSX R1 R2 (S is the size and X is a condition): if X is satisfied, move data from R1 to R2. R1, R2 cannot both be memory address.

## Example

movq %rdi, %rax subq %rsi, %rax movq %rsi, %rdx subq %rdi, %rdx cmpq %rsi, %rdi cmovge %rdx, %rax ret Logic Structures in Assembly Language

XD

ommands

Program Count Condition Code

est

Conditions

Set

Jump Tables

Conditiona Commands

Conditional Jumps

. . . . .

Logic Structures

Conditional Branches (if)
Loops (for, while)
Switch Statements (swite

## **Condition Moves**

So, why use cmov?

### Logic Structures in Assembly Language

XD

### Commands

Program Count Condition Code

Test

Condition

lumns

Jump Table

### Conditiona Command

Conditional Jump

### Conditional Moves

### ogic Structures

Conditional Branches (if

Switch Statements

- Advantages
  - Save space
  - Enhanced performance on pipelined CPUs (while jmp perform terribly) due to condition code transfers.

Logic Structures in Assembly Language

XD

ommands

Condition Codes

Гest

onditions

Jumps

Jump Tables

Commands

Conditional Moves

Logic Structures

Logic Structures

Loops (for, while) Switch Statements (swi

JWILLII JLALEIIIEII

Exercises

### Remark

Pipelined CPU achieves high performance by overlapping the steps of the successive instructions.



- Advantages
  - Save space
  - Enhanced performance on pipelined CPUs (while jmp perform terribly) due to condition code transfers.
- Disadvantages
  - May break the structure of assembly code
  - Increase the calculation workload
  - Forbidden in some cases

Logic Structures in Assembly Language

XD

Conditional Moves

### Remark

Pipelined CPU achieves high performance by overlapping the steps of the successive instructions.



Set

Jumps

Condition

Commands
Conditional Jum

Conditional Moves

I a mi a Chuu ahuuna

## Logic Structures

Loops (for, while)
Switch Statements (sw

Switch Statemen

Exercise

So, why use cmov?

- Advantages
  - Save space
  - Enhanced performance on pipelined CPUs (while jmp perform terribly) due to condition code transfers.
- Disadvantages
  - May break the structure of assembly code
  - Increase the calculation workload
  - ► Forbidden in some cases
- Bad cases
  - Expensive evaluations: function-involved
  - ► Risky computation: p ? \*p : 0
  - ► Computations with side effects: +=, \*=

### Remark

Pipelined CPU achieves high performance by overlapping the steps of the successive instructions.

### Conditional Branches (if)

## Implementation

cmov, conditional jumps

## Example

Implement a function that return abs of the first argument.

### Answer

mov %rdi %rax test %rax %rax ige .L1 sub \$0 %rax

.L1:

repz ret

## for, while

## **Implementation** conditional jumps

## Example

Implement a loop using %rdi and %rsi where the outer i iterates from 0 to 9 and the inner one iterates from i to 9 Logic Structures in Assembly Language

XD

Loops (for, while)

## for, while

## Answer mov \$0 %rdi jmp .CHECK1 .LOOP1: mov %rdi %rsi jmp .CHECK2 .LOOP2: ... //do stuff .CHECK2: cmp %rsi \$9 ile .LOOP2 .CHECK1: cmp %rdi \$9

ile .LOOP1

### Logic Structures in Assembly Language

XD

### Loops (for, while)

Switch Statements (switch)

## Implementation

jump tables

## Example

Implement a function on a 64x machine to jump to .L1 when %rax is prime and .L0 when it is not. (% $rax \in \{1, 2, 3, 4\}$ )

### Answer

.section .rodata

.align 8

.ANCHOR:

.quad .L0

.quad .L1

.quad .L1

.quad .L0

imp .ANCHOR(,%rax,\$8)

### Exercises

```
Implemented on a x86-64 AMD machine.
function f(int a, int b, char flag){
return flag?abs(a-b):-abs(a-b);
pushq %rbp
__(1)__ %dl %rdx
__(2)__ %rsi %rdi
__(3)__ %rbp
movzbg %bpl %rbp
__(4)__ %rbp %rdx
inz .L1
__(2)__ %rsi %rdi
_{-}(5)_{-} %rbp
movzbq %bpl %rbp
__(6)__ %rdx %rbp
__(7)__ .L1
mov %rdi %rax
sub __(8)__ __(9)__
```

### Logic Structures in Assembly Language

XD

### ommands

Program Counte Condition Codes Compare

Test Condition

Set

Jumps

Jump Tables

### Comman

Conditional Jum Conditional Mov

Logic Structures

### Conditional Branches

Loops (for, while) Switch Statements (switch

Exercises

Exercises

4 ロ ト 4 倒 ト 4 亘 ト 亘 め 9 0 0

## **Exercises**

```
.L1:
mov __(10)__ %rax
sub __(11)__ __(12)__
.RET:
pop %rbp
__(13)__ ret
```

### Logic Structures in Assembly Language

XD

### ommands

Program Counte Condition Codes Compare

Condition

Set

lumn Tables

Jump Tables

### Condition

Conditional Jump

### a alla Camination

Conditional Branches (if

Switch Statements